

# Mars 2020 (M2020)

## Software Interface Specification

Interface Title: **Camera Instrument Data Products**

Mission: M2020

Date: February 22, 2022

Module ID: D-99960

Module Type (REference Only or MISSION-specific info included): MIS

Reference Module ID: N/A

Date: N/A

### Signatures

#### ***GDS Generating Elements:***

Instrument Data Subsystem (IDS)  
Stirling Algermissen

\_\_\_\_\_  
Subsystem Lead

\_\_\_\_\_  
Date

#### ***Science Receiving Elements:***

M2020 Deputy Project Scientist  
Ken Williford

\_\_\_\_\_  
Deputy Project Scientist

\_\_\_\_\_  
Date

#### ***GDS Receiving Elements:***

M2020 GDS System Manager  
Amy Attiyah

\_\_\_\_\_  
System Manager

\_\_\_\_\_  
Date

M2020 GDS Chief Engineer  
Guy Pyrzak

\_\_\_\_\_  
System Engineer

\_\_\_\_\_  
Date

#### ***Concurrence:***

M2020 Imaging Investigation Scientist  
Justin Maki

\_\_\_\_\_  
Investigation Scientist

\_\_\_\_\_  
Date

M2020 Mastcam-Z Investigation Scientist  
Justin Maki

\_\_\_\_\_  
Investigation Scientist

\_\_\_\_\_  
Date

M2020 ECAM Instrument Ops Lead  
Nicholas A. Ruoff

\_\_\_\_\_  
Instrument Lead

\_\_\_\_\_  
Date

M2020 SuperCam Instrument Lead  
Dorothea Marcia Delapp

\_\_\_\_\_  
Instrument Lead

\_\_\_\_\_  
Date

M2020 MEDA Investigation Scientist  
Manuel de la Torre Juarez

\_\_\_\_\_  
Instrument Scientist

\_\_\_\_\_  
Date

M2020 MEDA Instrument Lead  
Luis Mora Sotomayor

Instrument Lead	Date
-----------------	------

M2020 SHERLOC Investigation Scientist  
Kevin Hand

Instrument Scientist	Date
----------------------	------

M2020 PIXL Investigation Scientist  
Yang Liu

Instrument Scientist	Date
----------------------	------

PDS Program Manager  
Tim McClanahan

Manager	Date
---------	------

PDS Geosciences Node Manager  
Ray Arvidson

Manager	Date
---------	------

PDS Discipline Node Manager  
Myche McAuley

Manager	Date
---------	------



# Mars 2020 Project

## *Software Interface Specification (SIS)*

### Camera Data Products

Version 3.1

**Prepared by:**

Nicholas A. Ruoff  
Robert G. Deen  
Oleg Pariser

**Custodian:**

Oleg Pariser

**JPL D-99960**



February 22, 2022

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

Jet Propulsion Laboratory, California Institute of Technology

© 2022 California Institute of Technology. Government sponsorship acknowledged.

## CHANGE LOG

DATE	SECTIONS CHANGED	REVISION
2019-11-17	Appendix B – updated for new IDPH (IMF) for all instruments and compression (JPEG) in FSW S6.0.1. Updated keyword sources for ZCAM and SHERLOC. Added new “FOCUS” keywords for ZCAM and SHERLOC.	0.7
2019-11-19	Appendix B – added ZCAM and SHERLOC valid values for COMMAND_INSTRUMENT_ID. Added PIXL Mini-Header Keywords	0.7
2020-02-20	Appendix B – updates for all instruments.	0.8
2020-04-30	Updated all tables and figures.	0.8
2020-05-04	Appendix B – updated keyword sources for PIXL, SCAM, ZCAM, and MEDA.	0.9
2020-05-06	Extensive reformatting. Consolidating of EDR info that spans all instruments.	0.9
2020-05-10	Appendix B – updates for telemetry sources. Add SHERLOC Engineering products (pass-through).	0.9
2020-5-15	All sections	0.9
2020-05-24	Appendix B - removed HDR group and placed HDR_* keywords back into OBSERVATION_REQUEST_PARMS. ACTIVITY_ID replaced keyword REQUEST_ID for all instruments.	1.0
2020-05-25	Updated “camera specific” portion of filename for SuperCam to reflect the “point number” of the RMI image. Updated time fields in filename for ZCAM/SHERLOC video. Added “RTT” and “PMC” to the RMC for PIXL. Updated the “Tool” CS group to reflect the name of the tool being used. Keyword OFFSET_MODE_ID changed to DC_OFFSET for all instruments.	1.0
2020-09-01	Reformatted Content Table Updated Signatorees list Added Tables and Figures Incorporated internal reviewers inputs Added Appendix C	1.0
2020-12-01	EDR peer-review feedback Added RDR sections	1.0
2020-12-21	RDR peer-review version	2.0
2021-01-04	Photogrammetric Camera Model Piluca’s additions and references Mastcam-Z operational Characteristics update based on Bell abstract for LPSC SCAM RMI Operational Char. update based on Gasnault abstract for LPSC Watson Operational Characteristics update based on Minitti abstract for LPSC ECAM Operational Characteristics update based on Maki abstract for LPSC	2.01
2021-01-12	Update to Arm Reachability States Table	2.01
2021-01-26	Fix compression table in the filename to reference MSSS	2.01
2021-04-07	Changed Naming convention for Colorglyphs to use “G”, instead of “C”	2.01
2021-07-29	Revisions to address PDS Peer Review Comments	3.0
2021-08-11	Release Version	3.01
2021-10-19	Updated RTE angular resolution and ACC FOV in characteristics tables	3.02
2021-10-27	Updated certain values in hazcam, ncaml, sherlock characteristics tables Updated names of pdf documents referenced in Appendix E and G	3.03

DATE	SECTIONS CHANGED	REVISION
2021-12-14	Updated Table 2-5 (RMI)	3.04
2022-01-24	Release 3 version: empty meshes in 9.1.2, xmlf change in 8.11 and 11.6; revise EDLcam and LCAM in 5.6.7, 5.6.8, 6.3.2; revise recovered images in 5.3.5; RMC/flight clarification in 4.4, 20.1, tables 18-1, -2, -3; heli extended ops in 14.2.2.4; heli idiosyncracies in 5.6.9.1; coord transform example in 20.11; cachecam flip in 5.6.1.2; camera characteristics in Appendix H; filename extensions in 3.2	3.1
2022-03-03	Section 8.5 Disparity added description of Disparity Coefficients	3.1

## Table of Contents

<b><u>LIST OF FIGURES.....</u></b>	<b><u>X</u></b>
<b><u>LIST OF TABLES .....</u></b>	<b><u>XII</u></b>
<b><u>ACRONYMS AND ABBREVIATIONS .....</u></b>	<b><u>XIII</u></b>
<b><u>REFERENCES .....</u></b>	<b><u>XX</u></b>
<b><u>1. INTRODUCTION.....</u></b>	<b><u>25</u></b>
1.1 PURPOSE AND SCOPE .....	25
1.2 HOW TO USE THIS DOCUMENT.....	26
1.3 CONSTRAINTS AND APPLICABLE DOCUMENTS .....	27
<b><u>2. INSTRUMENT OVERVIEW .....</u></b>	<b><u>28</u></b>
2.1 ENGINEERING CAMERA INSTRUMENT SUITE .....	30
2.1.1 HAZARD AVOIDANCE CAMERA (HAZCAM) .....	31
2.1.2 NAVIGATION CAMERA (NAVCAM).....	33
2.1.3 SAMPLE CACHING SYSTEM CAMERA (CACHECAM) .....	34
2.2 SUPERCAM REMOTE MICRO-IMAGER (RMI).....	36
2.3 MASTCAM-Z CAMERA INSTRUMENT SUITE .....	38
2.4 SHERLOC CAMERA SUITE .....	40
2.4.1 SHERLOC-WATSON .....	40
2.4.2 AUTOFOCUS AND CONTEXT IMAGER (ACI) .....	42
2.5 PIXL MICRO CONTEXT CAMERA (MCC) .....	43
2.6 MEDA SKY CAMERA.....	45
2.7 EDL CAMERA SUITE.....	46
2.7.1 PARACHUTE UPLOOK CAMERAS (PUC) .....	47
2.7.2 DESCENT STAGE DOWNLOOK CAMERA (DDC) .....	47
2.7.3 ROVER UPLOOK CAMERA (RUC) .....	47
2.7.4 ROVER DOWNLOOK CAMERA (RDC).....	48
2.7.5 EDL MICROPHONE.....	48
2.8 LANDER VISION SYSTEM CAMERA (LCAM) .....	48
2.9 HELICOPTER IMAGING SUITE .....	49
2.9.1 HELICOPTER NAVIGATION CAMERA .....	50
2.9.2 HELICOPTER RETURN TO EARTH CAMERA.....	51
2.10 CAMERA FLIGHT SOFTWARE PROCESSING .....	51
2.10.1 JPEG COMPRESSION.....	51
2.10.2 COLOR ICER FOR SUPERCAM.....	52
2.10.3 COMPRESSION FOR MEDA.....	52
<b><u>3. PRODUCT LABEL AND HEADER .....</u></b>	<b><u>53</u></b>

<b>3.1 ODL LABEL .....</b>	<b>53</b>
3.1.1 KEYWORD LENGTH LIMITS.....	54
3.1.2 DATA TYPE RESTRICTIONS.....	54
3.1.3 INTERPRETATION OF N/A, UNK, AND NULL .....	54
3.1.4 ODL LABEL CONSTRUCTS “CLASS”, “OBJECT” AND “GROUP” .....	55
3.1.5 ODL IMAGE OBJECT.....	56
<b>3.2 PDS4 LABEL .....</b>	<b>56</b>
<b>3.3 VICAR LABEL .....</b>	<b>56</b>
<b>3.4 MAPPING OF ODL AND VICAR LABELS .....</b>	<b>57</b>
<b>3.5 MAPPING OF PDS4 AND VICAR/ODL LABELS.....</b>	<b>59</b>
<b>3.6 LABEL GROUPS ORGANIZATION .....</b>	<b>61</b>
<b>3.7 VICAR/ODL LABEL MAP.....</b>	<b>62</b>
<b>3.8 PDS4 LABEL MAP.....</b>	<b>63</b>
<b>3.9 SUPERCAM FITS FILES .....</b>	<b>65</b>

#### **4. GENERAL DATA PRODUCT OVERVIEW..... 66**

<b>4.1 DATA PROCESSING LEVELS.....</b>	<b>66</b>
<b>4.2 BINARY DATA STORAGE CONVENTIONS .....</b>	<b>67</b>
4.2.1 BIT AND BYTE ORDERING.....	67
<b>4.3 TIME CONVENTIONS.....</b>	<b>67</b>
<b>4.4 ROVER MOTION COUNTER (RMC) .....</b>	<b>68</b>
4.4.1 ROVER SURFACE OPS RMC NAMESPACE.....	69
4.4.2 HELICOPTER RMC NAMESPACE .....	70
4.4.3 LCAM RMC NAMESPACE .....	71
4.4.4 RMC AND COORDINATE FRAMES .....	71

#### **5. EDR PRODUCT SPECIFICATION..... 72**

<b>5.1 EDR GENERAL PROCESSING.....</b>	<b>72</b>
<b>5.2 IMAGE EDR SUB-CATEGORIES .....</b>	<b>74</b>
5.2.1 RAW EDRS .....	74
5.2.2 DECOMPANDED EDRS .....	75
5.2.3 DEBAYERED EDRS .....	75
<b>5.3 EDR IMAGE TYPES.....</b>	<b>76</b>
5.3.1 STANDARD IMAGES .....	77
5.3.2 Z-STACKS (BEST FOCUS IMAGES).....	77
5.3.3 DEPTH MAPS .....	78
5.3.4 VIDEO FRAMES.....	79
5.3.5 RECOVERED IMAGES .....	79
5.3.6 RAW JPEG EDR (EJP).....	80
<b>5.4 EDR IMAGE SAMPLING .....</b>	<b>81</b>
5.4.1 SUBFRAMING .....	81
5.4.2 DOWNSAMPLING.....	81
5.4.3 PIXEL BINNING.....	81
5.4.4 CO-ADDING .....	82
5.4.5 THUMBNAILS .....	82
<b>5.5 EDR PRODUCT FORMAT .....</b>	<b>83</b>
<b>5.6 INSTRUMENT-SPECIFIC IMAGE EDR DETAILS .....</b>	<b>85</b>

5.6.1 ENGINEERING CAMERA INSTRUMENT SUITE EDRs .....	85
5.6.2 SUPERCAM RMI EDRs.....	88
5.6.3 MASTCAM-Z (ZCAM) EDRs.....	89
5.6.4 SHERLOC CAMERA INSTRUMENT SUITE EDRs.....	90
5.6.5 PIXL MCC EDRs.....	90
5.6.6 MEDA SKYCAM .....	90
5.6.7 EDL CAMERA SUITE .....	90
5.6.8 LCAM .....	92
5.6.9 HELICOPTER .....	92
<b>5.7 IMAGE SUPPORT EDRs.....</b>	<b>93</b>
5.7.1 Row SUMMATION EDR (ERS).....	93
5.7.2 COLUMN SUMMATION EDR (ECS) .....	93
5.7.3 REFERENCE PIXEL EDR (ERP) .....	93
5.7.4 HISTOGRAM EDR (EHG).....	94
5.7.5 IDPH ONLY (EID) .....	94
<b>5.8 ENGINEERING EDRs.....</b>	<b>94</b>
5.8.1 SUPERCAM RMI.....	94
5.8.2 PIXL-MCC.....	94
5.8.3 SHERLOC .....	95
5.8.4 NAVMAP EDRs.....	96
<b>5.9 OTHER EDRs .....</b>	<b>96</b>
5.9.1 MOVIE EDRs .....	96
5.9.2 EDL MICROPHONE.....	96
<b>5.10 EDR PRODUCT VALIDATION.....</b>	<b>96</b>

## **6. FDR PRODUCT SPECIFICATION..... 98**

<b>6.1 FDR GENERAL PROCESSING.....</b>	<b>98</b>
<b>6.2 ECAM FDR PROCESSING (RECONSTRUCTED IMAGES) .....</b>	<b>100</b>
6.2.1 TILE RECONSTRUCTION TYPES.....	100
6.2.2 ECAM FUNDAMENTAL TILE DATA RECORD (TDR).....	102
<b>6.3 LABEL UPDATES.....</b>	<b>102</b>
6.3.1 PIXL LABEL UPDATES.....	102
6.3.2 LCAM LABEL UPDATES .....	102
<b>6.4 AUDIO DATA.....</b>	<b>103</b>

## **7. PIXEL-BASED IMAGE PROCESSING..... 104**

<b>7.1 RADIOMETRIC CORRECTION.....</b>	<b>104</b>
7.1.1 MIPLRAD .....	104
7.1.2 CHEMRAD .....	108
7.1.3 MEDA SKYCAM RADIOMETRIC PROCESSING .....	108
7.1.4 ZCAM RADIOMETRIC PROCESSING.....	109
7.1.5 RADIOMETRIC FLOAT TO INTEGER SCALING .....	110
7.1.6 ZENITH SCALING.....	111
<b>7.2 COLOR CORRECTION .....</b>	<b>112</b>
<b>7.3 GAMMA CORRECTION .....</b>	<b>114</b>
<b>7.4 PHOTOMETRY .....</b>	<b>115</b>

<b>8. GEOMETRIC AND STEREO PROCESSING.....</b>	<b>116</b>
<b>8.1 STEREO PAIR MATCHING.....</b>	<b>116</b>
<b>8.2 CAMERA MODELS.....</b>	<b>116</b>
8.2.1 CAHV FAMILY OF CAMERA MODELS .....	117
8.2.2 PSPH CAMERA MODEL .....	118
8.2.3 PHOTOGRAMMETRIC CAMERA MODEL.....	119
8.2.4 INTERPOLATING CAMERA MODELS.....	120
<b>8.3 POINTING A CAMERA MODEL.....</b>	<b>121</b>
<b>8.4 EPIPOLAR ALIGNMENT AND LINEARIZATION.....</b>	<b>123</b>
<b>8.5 DISPARITY .....</b>	<b>124</b>
<b>8.6 CORRELATION .....</b>	<b>125</b>
<b>8.7 XYZ GENERATION .....</b>	<b>126</b>
8.7.1 ORBITAL XYZ.....	128
<b>8.8 RANGE ERROR.....</b>	<b>129</b>
<b>8.9 RANGE FILTERING.....</b>	<b>130</b>
8.9.1 RANGE FILTER ALGORITHM DETAILS.....	131
<b>8.10 RANGE.....</b>	<b>133</b>
<b>8.11 ROVER MASK.....</b>	<b>133</b>
<b>8.12 SURFACE NORMAL .....</b>	<b>134</b>
<b>8.13 SLOPE MAPS.....</b>	<b>134</b>
8.13.1 SLOPE (NOMINAL) .....	134
8.13.2 SLOPE HEADING .....	134
8.13.3 SLOPE ROVER DIRECTION .....	135
8.13.4 SLOPE MAGNITUDE .....	135
8.13.5 SLOPE NORTHERLY TILT.....	135
<b>8.14 ANAGLYPH/COLORGLYPH.....</b>	<b>135</b>
<b>8.15 PIXEL SCALE FOR SINGLE-FRAME IMAGES.....</b>	<b>136</b>
 <b>9. 3D SURFACE RECONSTRUCTION AND MESHES.....</b>	 <b>137</b>
<b>9.1 TACTICAL TERRAIN MESH.....</b>	<b>137</b>
9.1.1 ADAPTIVE MESH POLYGON SIZING .....	137
9.1.2 FORMAT DESCRIPTION .....	137
9.1.3 UNIFIED TACTICAL TERRAIN MESH.....	138
<b>9.2 ORBITAL DEM-BASED MESH .....</b>	<b>138</b>
<b>9.3 CONTEXTUAL MESHES .....</b>	<b>139</b>
 <b>10. CONTACT SCIENCE PROCESSING.....</b>	 <b>142</b>
<b>10.1 SURFACE ROUGHNESS.....</b>	<b>143</b>
<b>10.2 ARM REACHABILITY .....</b>	<b>144</b>
<b>10.3 GOODNESS MAPS .....</b>	<b>145</b>
 <b>11. MOSAIC GENERATION.....</b>	 <b>147</b>
<b>11.1 OVERVIEW OF MOSAICS.....</b>	<b>147</b>
<b>11.2 POINTING CORRECTION.....</b>	<b>147</b>

<b>11.3 BRIGHTNESS CORRECTION .....</b>	<b>148</b>
<b>11.4 MOSAICKING METHOD FOR MOST MOSAICS.....</b>	<b>149</b>
11.4.1 OUTPUT PROJECTION DETERMINATION.....	149
11.4.2 SURFACE DETERMINATION .....	149
11.4.3 COMPUTATION OF OUTPUT VIEW RAY.....	149
11.4.4 PROJECTION FROM OUTPUT TO SURFACE .....	150
11.4.5 PROJECTION FROM SURFACE TO INPUT .....	150
11.4.6 INTERPOLATION AND STORAGE OF THE RESULT .....	150
<b>11.5 MOSAICKING METHOD FOR ORTHORECTIFIED MOSAICS .....</b>	<b>150</b>
<b>11.6 MOSAIC ANCILLARY FILES.....</b>	<b>151</b>
 <b><u>12. MOSAIC PROJECTIONS.....</u></b>	 <b><u>154</u></b>
<b>12.1 CYLINDRICAL PROJECTION .....</b>	<b>154</b>
<b>12.2 CAMERA POINT PERSPECTIVE.....</b>	<b>154</b>
<b>12.3 CYLINDRICAL-PERSPECTIVE.....</b>	<b>155</b>
<b>12.4 POLAR PROJECTION .....</b>	<b>157</b>
<b>12.5 VERTICAL PROJECTION.....</b>	<b>158</b>
<b>12.6 ORTHOGRAPHIC PROJECTION .....</b>	<b>159</b>
<b>12.7 ORTHORECTIFIED PROJECTION .....</b>	<b>160</b>
<b>12.8 NON-IMAGE MOSAICS .....</b>	<b>162</b>
<b>12.9 MOSAIC INDEX FILES (ICM/IDX) .....</b>	<b>162</b>
<b>12.10 PIXEL SCALE FOR MOSAICS .....</b>	<b>162</b>
 <b><u>13. OTHER PROCESSING .....</u></b>	 <b><u>164</u></b>
<b>13.1 TERRAIN CLASSIFICATION.....</b>	<b>164</b>
<b>13.2 INCIDENCE, EMISSION, PHASE.....</b>	<b>164</b>
<b>13.3 IMAGE OVERLAYS.....</b>	<b>164</b>
<b>13.4 FILLED IMAGES.....</b>	<b>164</b>
 <b><u>14. INSTRUMENT-SPECIFIC PROCESSING.....</u></b>	 <b><u>165</u></b>
<b>14.1 MASTCAM-Z .....</b>	<b>165</b>
<b>14.2 HELICOPTER.....</b>	<b>166</b>
14.2.1 HELICOPTER IMAGES.....	166
14.2.2 HELICOPTER PLACEMENT SUPPORT .....	166
<b>14.3 PIXL MCC TRAPEZOID CORRECTION.....</b>	<b>170</b>
<b>14.4 MEDA SKYCAM.....</b>	<b>171</b>
<b>14.5 CRUISE FLIGHT SOFTWARE IDIOSYNCRACIES .....</b>	<b>171</b>
 <b><u>15. COREGISTRATION AND LOCALIZATION .....</u></b>	 <b><u>173</u></b>
<b>15.1 IMAGE COREGISTRATION.....</b>	<b>173</b>
15.1.1 ICM PRODUCTS.....	173
15.1.2 ICM FILENAME.....	175
15.1.3 PIPELINE ICM GENERATION .....	176



15.1.4 CRISP SERVER .....	176
15.1.5 CRUST SERVER .....	176
<b>15.2 LOCALIZATION .....</b>	<b>177</b>
15.2.1 ONBOARD LOCALIZATION .....	177
15.2.2 AUTO-LOCALIZATION.....	177
15.2.3 MANUAL LOCALIZATION.....	178
15.2.4 SCIENCE LOCALIZATION .....	178
<b>15.3 PLACES .....</b>	<b>178</b>
<b>15.4 C-POSE .....</b>	<b>179</b>
 <b><u>16. AUTOMATION AND DISTRIBUTION .....</u></b>	 <b><u>180</u></b>
 16.1 CLOUD ARCHITECTURE.....	180
16.2 MISSION DATA FLOW .....	182
16.3 PIPELINE AUTOMATION .....	182
16.4 DATA DISTRIBUTION.....	184
 <b><u>17. PRODUCT TYPE CODES .....</u></b>	 <b><u>185</u></b>
 <b><u>18. FILE NAMING STANDARDS.....</u></b>	 <b><u>192</u></b>
 18.1 EDR/RDR SINGLE-FRAME FILENAME.....	193
18.2 MOSAIC RDR FILENAME .....	206
18.3 UNIFIED TERRAIN MESH RDR FILENAME.....	214
 <b><u>19. FILENAME MATCHING AND VERSIONING.....</u></b>	 <b><u>220</u></b>
 19.1 VERSION-LIKE FIELDS .....	220
19.2 DETERMINING INSTANCES OF THE “SAME” IMAGE .....	220
19.3 SORT ORDER FOR VERSION-LIKE FIELDS .....	221
19.4 VERSION CONSISTENCY BETWEEN OPS AND PDS .....	223
 <b><u>20. COORDINATE FRAME STANDARDS .....</u></b>	 <b><u>224</u></b>
 20.1 COORDINATE FRAME INDICES .....	225
20.2 ROVER NAVIGATION (ROVER NAV) FRAME.....	225
20.3 ROVER MECHANICAL (ROVER MECH) FRAME .....	227
20.4 LOCAL LEVEL FRAME .....	227
20.5 SITE FRAME.....	227
20.6 RSM FRAME.....	228
20.7 ARM FRAMES.....	228
20.8 PIXL FRAMES.....	229
20.8.1 PIXL_BASE FRAME .....	229
20.8.2 PIXL_TOOL FRAME.....	229
20.8.3 PIXL_SENSOR FRAME.....	229
20.8.4 PIXL_CAMERA FRAME .....	229
20.9 HELICOPTER FRAMES .....	230

20.9.1 HELICOPTER MECHANICAL FRAME (“M” OR “CSYS2”).....	232
20.9.2 HELICOPTER IMU FRAME (“S1” AND “S2”) .....	232
20.9.3 HELICOPTER GROUND FRAME (“G”) .....	232
<b>20.10 LANDER VISION SYSTEM CAMERA FRAMES.....</b>	<b>233</b>
20.10.1 MARS CENTERED MARS FIXED FRAME (“MCMF”) .....	233
20.10.2 LCAM INTERNAL FRAME (“CINT”).....	233
<b>20.11 EXAMPLE COORDINATE FRAME TRANSFORM .....</b>	<b>234</b>
 <b><u>21. APPLICABLE SOFTWARE .....</u></b>	 <b><u>240</u></b>
 21.1 UTILITY PROGRAMS.....	240
21.2 APPLICABLE PDS SOFTWARE TOOLS .....	241
21.3 SOFTWARE DISTRIBUTION AND UPDATE PROCEDURES.....	241
 <b><u>APPENDIX A. DATA PRODUCT AND DATA PRODUCT OBJECT NAMES .....</u></b>	 <b><u>242</u></b>
 <b><u>APPENDIX B. PRODUCT LABEL KEYWORD DEFINITIONS, VALUES, AND SOURCES .....</u></b>	 <b><u>247</u></b>
 <b><u>APPENDIX C. CALIBRATION PARAMETERS.....</u></b>	 <b><u>250</u></b>
 <b><u>APPENDIX D. PRODUCT FLOW DIAGRAM .....</u></b>	 <b><u>253</u></b>
 <b><u>APPENDIX E. SHERLOC WATSON AND ACI [SUPPLEMENTARY INFORMATION TABLE] .....</u></b>	 <b><u>254</u></b>
 <b><u>APPENDIX F. SCLK USAGE IN THE PRODUCTS FILENAMES AND THE METADATA .....</u></b>	 <b><u>257</u></b>
 <b><u>APPENDIX G. SPECIAL PRODUCT FLAG TABLE .....</u></b>	 <b><u>258</u></b>
 <b><u>APPENDIX H. CAMERA CHARACTERISTICS TABLES .....</u></b>	 <b><u>259</u></b>

## LIST OF FIGURES

Figure 2-1: Example Bayer filter pattern .....	29
Figure 2-2: The cameras on M2020 Rover .....	29
Figure 2-3: Engineering Camera hardware design .....	30
Figure 2-4: Front Hazcam locations .....	32
Figure 2-5: Front Hazcam stereo pair mapping. Blue is the A-side pair, red is the B-side pair. ....	32
Figure 2-6: Rear Hazcam location .....	33
Figure 2-7: Navcam location .....	34
Figure 2-8: CacheCam location (yellow cylinder in Vision Assessment Station) .....	35
Figure 2-9: SuperCam Mast Unit .....	36
Figure 2-10: RMI Image Data Formats .....	37
Figure 2-11: Mastcam-Z Spectral Response .....	40
Figure 2-12: WATSON("Wide Angle Imager") and ACI("Context Imager") Cameras of SHERLOC instrument suite. ....	42
Figure 2-13: Images of the MCC components and location of sensor head components within the PIXL Sensor Assembly .....	44
Figure 2-14: The PIXL Sensor Assembly .....	44
Figure 2-15: LED distribution on the FLI and corresponding LED ID .....	45
Figure 2-16: RDS-Skycam with FOV .....	46
Figure 2-17: EDL Cameras overview .....	47
Figure 2-18: Helicopter Cameras Photo. Bottom View: RTE is at the lower corner in the indentation. Navigation Camera is the "nose" in the oval cutout .....	50
Figure 2-19: Helicopter Cameras location and viewing frustrums. Navigation Camera looking straight down, RTE angled. ....	51
Figure 3-1: Product Types that have detached PDS4 label .....	60
Figure 5-1: Instrument Data Flow to IDS .....	73
Figure 5-2: IDS EDR generation data flow .....	74
Figure 5-3: An example of a "zipper" pattern: image processing artifact introduced by the Malvar de- mosaicking algorithm (MSL Mastcam M100 Sol 1155) .....	76
Figure 5-4: ECAM multiple modes of pixel binning .....	82
Figure 5-5: CacheCam rotation. Left: original EJP. Right: raw EDR after flip .....	88
Figure 5-6: RMI Rotation. Left: original EJP. Right: raw EDR after rotation and flip .....	88
Figure 5-7: ZCAM – RGB in Bayer Pattern Layout on CCD .....	89
Figure 6-1: Flow Diagram for different paths to generate FDR .....	99
Figure 6-2: ECAM Example of single downsampling resolution. ....	101
Figure 7-1: RMI Image Pipeline (simplified) .....	108
Figure 7-2 High level flow chart describing radiance calibration of Skycam images. ....	109
Figure 7-3: High-level flow chart describing radiance ("RAD") and radiance factor ("IOF") calibration of raw Mastcam-Z EDRs, conducted at Arizona State University based on Mastcam-Z instrument and calibration target performance and characteristics described in Bell et al. (2020), Hayes et al. (2020), and Kinch et al. (2020). ....	110
Figure 7-4: Color and Radiometric Processing Flow. The code contains the 3-letter product type and the Filter/Color code (indicating the color type) .....	114
Figure 8-1: Example XYZ image overlay from MSL .....	128
Figure 8-2: Range Error to 10m for M2020 Cameras .....	130
Figure 8-3: Range Error to 75m for M2020 Cameras .....	130
Figure 9-1: Robotic Arm workspace is rendered in Tactical Terrain Mesh and visualized in RSVP HyperDrive .....	138
Figure 9-2: Orbital terrain mesh of M2020 Landing Site in Jezero crater as visualized in RSVP Hyperdrive. ....	139

Figure 9-3: Contextual mesh in first person perspective (top) and third person perspective (bottom).	140
Figure 10-1: Contact Science Product Flow .....	142
Figure 10-2: Drill Topography Requirements.....	143
Figure 10-3: Relative Radii of Roughness Products.....	144
Figure 12-1: Cylindrical Projection.....	154
Figure 12-2: Point Perspective Projection .....	155
Figure 12-3: Cylindrical-Perspective Projection.....	157
Figure 12-4: Polar projection .....	158
Figure 12-5: Vertical Projection .....	159
Figure 12-6: Orthorectified Projection.....	161
Figure 14-1: Mastcam-Z Stereo Coverage .....	165
Figure 14-2: Helicopter Flight Zone .....	167
Figure 14-3: Example of PIXL Trapezoidal Correction .....	171
Figure 15-1: Coregistration example .....	173
Figure 16-1: Mission Operations Architecture .....	180
Figure 16-2: A screenshot of the Campaign Analysis Mapping and Planning (CAMP) Application .....	181
Figure 16-3: PDS4 Label Generation using APPS .....	182
Figure 16-4: Data Flow diagram from acquisition on the spacecraft to processing by IDS pipeline.....	182
Figure 16-5: Typical BPMN Diagram for an IDS Pipeline product generation Flow .....	183
Figure 16-6: The IDS Pipeline Console web application .....	184
Figure 20-1: Rover navigation coordinate frame (RNAV).....	226
Figure 20-2: Yaw, Pitch and Roll Definition .....	227
Figure 20-3: Site and Rover frame examples .....	228
Figure 20-4: PIXL Coordinate Frames.....	230
Figure 20-5: Helicopter M(aka Csys2) Frame.....	231
Figure 20-6: Mars Centered Mars Fixed.....	233
Figure 20-7: LCAM internal frame (Cint).....	233
Figure 20-8: Sample helicopter image from Flight 12 (HNM_0174_0682390538_578FDR_N0120001HELI01649_0000LUJ01.IMG). Left side is original, right side rotated 164.2 degrees clockwise (see text). .....	234
Figure 20-9: Orbital map of helicopter location (cutout from base map).....	237
Figure 21-1: Simplified Schematics of core RDR products generation.....	253

## LIST OF TABLES

Table 2-1 – Tabulation of M2020 Camera Instruments .....	28
Table 2-2 – Hazcam operational characteristics.....	31
Table 2-3 – Navcam Operational Characteristics .....	33
Table 2-4 – CacheCam Operational Characteristics .....	35
Table 2-5 – SCAM RMI Operational Characteristics .....	36
Table 2-6 – RMI Compression Modes .....	37
Table 2-7 – Mastcam-Z Operational Characteristics .....	38
Table 2-8 – Mastcam-Z Spectral Filters and Bandpass.....	39
Table 2-9 – ZCAM Focal Length to Zoom Motor Counts.....	40
Table 2-10 – SHERLOC-WATSON Operational Characteristics.....	41
Table 2-11 – SHERLOC-ACI Operational Characteristics.....	42
Table 2-12 – PIXL-MCC Operational Characteristics .....	45
Table 2-13 – SkyCam Operational Characteristics.....	46
Table 2-14 – EDL Cameras Operational Characteristics.....	48
Table 2-15 – LCAM Operational Characteristics .....	48
Table 2-16 – Helicopter Cameras Operational Characteristics .....	49
Table 3-1 – ODL Class to VICAR Property Set Mappings.....	58
Table 4-1 – PDS4 Processing Levels for Instrument Experiment Data Sets.....	66
Table 4-2 – M2020 Image EDR/RDR Bit Ordering .....	67
Table 4-3 – RMC Namespaces .....	69
Table 5-1: Relation between commanded image participant in an onboard SHERLOC WATSON,ACI or Mastcam-Z focus merge and depth map grayscale data value (DN) [Edgett2019].....	79
Table 5-2 – Type Codes for Image EDRs.....	83
Table 5-3 – List of EDR Types and Formats .....	83
Table 5-4 – DC Offset specification for the Engineering Cameras .....	87
Table 5-5 – Bayer pattern interpolation per filter for Mastcam-Z .....	89
Table 8-1 – Range Error Parameters .....	129
Table 8-2 – Range Error vs Cross-Range Resolution (Mastcam-Z at 34mm).....	130
Table 10-1 – Roughness Status Band Values.....	144
Table 10-2 – Arm Instruments .....	145
Table 10-3 – Arm reachability bit assignments per configuration .....	145
Table 10-4 – Arm reachability values .....	145
Table 10-5 – Goodness Value Interpretation.....	146
Table 14-1 – Helicopter Placement Product Bands.....	168
Table 14-2 – Helicopter Goodness Values .....	169
Table 17-1 – Product Type color codes.....	185
Table 17-2 – M2020 Product Type Codes.....	185
Table 17-3 – M2020 Product Type Codes, Sorted Alphabetically .....	189
Table 18-1 – Camera instrument single-frame filenaming convention .....	193
Table 18-2 – Camera instrument mosaic filenaming convention.....	206
Table 18-3 – Camera instrument unified terrain mesh filenaming convention .....	214
Table 20-1 – Coordinate frames used for M2020 surface operations .....	224
Table 21-1 – Key software toolsets .....	240

## ACRONYMS AND ABBREVIATIONS

Note that there are a number of 3-letter product type codes that are used throughout this document. These are listed in Section 17 and are not repeated in this table.

<b>3D</b>	Three-Dimensional
<b>3PCS</b>	Pointing, Positioning, Phasing, and Coordinate Systems
<b>ACA</b>	Adaptive Caching Assembly
<b>ACI</b>	Autofocus and Context Imager (SHERLOC)
<b>AF</b>	Autofocus
<b>AMPCS</b>	Advanced Multi-mission data Processing and Control Subsystem
<b>APID</b>	Application Process Identifier
<b>APSS</b>	Activity Planning and Sequencing Subsystem
<b>ASCII</b>	American Standard Code for Information Interchange
<b>ASTTRO</b>	Advanced Science Targeting Tool for Robotic Operations
<b>ATLO</b>	Assembly, Test, Launch and Operations
<b>BIST</b>	Built-In Self Test (SHERLOC)
<b>BITCAR</b>	Bit-Carousel
<b>BPM</b>	Business Processing Management
<b>BPMN</b>	Business Process Model and Notation
<b>BU</b>	Body-Unit (SuperCam)
<b>CacheCam</b>	Sample Caching System Camera
<b>CAHV</b>	Center, Axis, Horizontal, Vertical (camera model)
<b>CAHVOR</b>	Center, Axis, Horizontal, Vertical, Optical, Radial (camera model)
<b>CAHVORE</b>	Center, Axis, Horizontal, Vertical, Optical, Radial, Entrance (camera model)
<b>CAMP</b>	Campaign Analysis Mapping and Planning
<b>CBLO</b>	Core Bit Lock Out
<b>CC</b>	Circle Centroiding (PIXL)
<b>CCD</b>	Charged Coupled Device
<b>CCBU</b>	Chemistry Camera Body Unit
<b>CCMU</b>	Chemistry Camera Mast Unit
<b>CEDL</b>	Cruise Entry Descent and Landing
<b>CHEMRAD</b>	SuperCam Radiometric Processing Algorithm
<b>CHU</b>	Camera Head Unit (PIXL)
<b>CIE</b>	Commission Internationale de l'Eclairage
<b>CMOS</b>	Complementary Metal Oxide Semiconductor
<b>CNES</b>	Centre National d'Etudes spatiales (French Space Agency)

<b>CODMAC</b>	Committee on Data Management and Computation
<b>COTS</b>	Commercial Off The Shelf
<b>C-POSE</b>	Camera Positions and Orientations in Surface Environments
<b>CSV</b>	Comma-separated-value
<b>CWS</b>	Common Workflow Service
<b>DDC</b>	Descent Stage Downlook Camera (EDLcam)
<b>DEA</b>	Digital Electronics Assembly
<b>DEM</b>	Digital Elevation Map (or Model)
<b>DIMU</b>	Descent Inertial Measurement Unit (a Honeywell MIMU)
<b>DN</b>	Digital Number
<b>DOY</b>	Day of Year
<b>DP</b>	Data Product (telemetry)
<b>DPO</b>	Data Product Object (telemetry)
<b>DPU</b>	Data Processing Unit (PIXL)
<b>DTE</b>	Direct to Earth
<b>DVT</b>	Data Validity Time
<b>ECAM</b>	Engineering Camera
<b>EECAM</b>	Enhanced Engineering Camera
<b>EDL</b>	Entry, Descent, and Landing
<b>EDLcam</b>	Entry, Descent, and Landing Camera Suite
<b>EDR</b>	Experiment Data Record
<b>EHA</b>	Engineering, Housekeeping & Accountability (EH&A)
<b>EM</b>	Engineering Model
<b>EMD</b>	Earth Metadata file (“emd”)
<b>EOL</b>	End Of Label
<b>EPDU</b>	End-of-Product PDU
<b>ERT</b>	Earth Received Time
<b>FDD</b>	Functional Design Document
<b>FDR</b>	Fundamental Data Record
<b>FEI</b>	File Exchange Interface
<b>FGICD</b>	Flight-Ground ICD
<b>FHAZ</b>	Front Hazard Avoidance Camera
<b>FITS</b>	Flexible Image Transport System
<b>FLAC</b>	Free Lossless Audio Codec
<b>FLI</b>	Flood Light Illuminator (PIXL)
<b>FLIR</b>	Vendor (company) name
<b>FM</b>	Flight Model

This document has been reviewed and determined not to contain export controlled technical data.

<b>FOV</b>	Field of View
<b>FPGA</b>	Field Programmable Gate Array
<b>FSW</b>	Flight Software
<b>GIS</b>	Geographic Information System
<b>GNC</b>	Guidance, Navigation, and Control
<b>GOP</b>	Group of Pictures
<b>GSFC</b>	Goddard Space Flight Center
<b>GDS</b>	Ground Data System
<b>GSE</b>	Ground Support Equipment
<b>Hazcam</b>	Hazard Avoidance Camera
<b>HDU</b>	Header Data Unit (FITS)
<b>HGA</b>	High Gain Antenna
<b>HWHM</b>	Half Width Half Maximum
<b>IC</b>	Inlets Cover
<b>ICD</b>	Interface Control Document
<b>ICER</b>	Image compression algorithm (not an acronym)
<b>ID</b>	Identification
<b>IDL</b>	Interactive Data Language
<b>IDPH</b>	Instrument Data Product Header
<b>IDS</b>	Instrument Data Subsystem
<b>IEC</b>	International Electrotechnical Commission
<b>I/F</b>	Incidence/Flux
<b>IFOV</b>	Instantaneous Field of View
<b>IK</b>	Inverse Kinematics (Robotic Arm)
<b>ILUT</b>	Inverse Lookup Table
<b>IMU</b>	Inertial Measurement Unit
<b>InSight</b>	Interior Exploration using Seismic Investigations, Geodesy, and Heat Transport
<b>IPE</b>	Integrated Planning and Execution (MS element)
<b>IRAP</b>	Institut de Recherche en Astrophysique et Planétologie
<b>ISIS</b>	Integrated Software for Imagers and Spectrometers
<b>IVP</b>	Inertial Vector Propagation
<b>JMS</b>	Java Message Server
<b>JPEG</b>	Joint Photographic Experts Group (compression)
<b>JPL</b>	Jet Propulsion Laboratory
<b>LANL</b>	Los Alamos National Laboratory
<b>LCAM</b>	Lander Vision System Camera
<b>LIBS</b>	Laser-Induced Breakdown Spectrometer (SuperCam)

This document has been reviewed and determined not to contain export controlled technical data.



<b>LOCO</b>	LOw-COMplexity, LOssless COMpression
<b>LOD</b>	Levels of Detail (Meshes)
<b>LUT</b>	LookUp Table
<b>LVS</b>	Lander Vision System
<b>MAHLI</b>	Mars Hand Lens Imager (MSL)
<b>MARDI</b>	Mars Descent Imager (MSL)
<b>Mastcam-Z</b>	Mast camera-Zoom
<b>MCC</b>	Micro Context Camera (PIXL)
<b>MCZ</b>	Mastcam-Z ( <b>obsolete: use ZCAM</b> )
<b>MEDA</b>	Mars Environmental Dynamics Analyzer
<b>MER</b>	Mars Exploration Rover
<b>MIC</b>	Microphone
<b>MIPL</b>	Multimission Instrument Processing Laboratory
<b>MIPLRAD</b>	MIPL Radiometric Correction
<b>MOS</b>	Mission Operations System
<b>MPCS</b>	Mission data Processing and Control Subsystem
<b>MPDU</b>	Metadata Protocol Data Unit
<b>MPEG</b>	Moving Picture Experts Group (video compression)
<b>MPF</b>	Mars Pathfinder
<b>MS</b>	Mission System
<b>MSL</b>	Mars Science Labortory
<b>MSSS</b>	Malin Space Science Systems
<b>MU</b>	Mast Unit (SuperCam)
<b>M2020</b>	Mars 2020 Rover (or Mission)
<b>N/A</b>	Not Applicable
<b>NAIF</b>	Navigation and Ancillary Information Facility
<b>NASA</b>	National Aeronautics and Space Administration
<b>Nav</b>	Navigation Camera (Helicopter)
<b>Navcam</b>	Navigation Camera (Rover)
<b>NVM</b>	Non-Volatile Memory
<b>OCS</b>	Operations Cloud Store
<b>ODS</b>	Operational Data Store
<b>ODL</b>	Object Description Language
<b>OPGS</b>	Operations Products Generation Subsystem
<b>PDC</b>	Parachute Downlook Camera
<b>PDS</b>	Planetary Data System

<b>PDS4</b>	Planetary Data System, Version 4
<b>PDU</b>	Protocol Data Unit
<b>PI</b>	Principal Investigator
<b>PIXL</b>	Planetary Instrument for X-ray Lithochemistry
<b>PLACES</b>	Position Localization and Attitude Correction Estimate Storage
<b>PMC</b>	PIXL Motion Counter
<b>PNG</b>	Portable Network Graphics
<b>PPDU</b>	Product Data Protocol Data Unit
<b>PRT</b>	Platinum Resistance Thermometer
<b>PSDD</b>	Planetary Science Data Dictionary
<b>PSPH</b>	Planospheric (camera model)
<b>PUC</b>	Parachute Uplook Camera (EDLcam)
<b>RA</b>	Robotic Arm
<b>RCE</b>	Rover Compute Element
<b>RDC</b>	Rover Downlook Camera (EDLcam)
<b>RDR</b>	Reduced Data Record
<b>RDS</b>	Amazon Rational Databases
<b>RGB</b>	Red, Green, Blue
<b>RGGB</b>	Red, Green, Green, Blue (Bayer pattern)
<b>RHAZ</b>	Rear Hazard Avoidance Camera
<b>RIMU</b>	Rover Inertial Measurement Unit (an Northrup-Grumman LN-200S)
<b>RMI</b>	Remote Micro-Imager (SuperCam)
<b>ROI</b>	Region of Interest
<b>ROLO</b>	ROver Localization
<b>RPS</b>	Rover Planning System
<b>RSM</b>	Remote Sensing Mast
<b>RTE</b>	Return To Earth camera (Helicopter)
<b>RTO</b>	Real Time Operations (MS element)
<b>RTT</b>	Round Trip Tracking
<b>RUC</b>	Rover Uplook Camera (EDLcam)
<b>RSVP</b>	Rover Sequencing and Visualization Program
<b>S3</b>	Amazon Simple Storage Server
<b>SAPP</b>	Surface Attitude, Positioning and Pointing
<b>SCAM</b>	SuperCam instrument
<b>SCID</b>	Spacecraft ID
<b>SCLK</b>	Spacecraft Clock
<b>SCM</b>	Spacecraft Configuration Manager

This document has been reviewed and determined not to contain export controlled technical data.

<b>SCS</b>	Sample Cache System
<b>SFDU</b>	Standard Format Data Unit
<b>SHA</b>	Sample Handling Arm
<b>SHERLOC</b>	Scanning Habitable Environments with Raman and Luminescence for Organics and Chemicals
<b>SIS</b>	Software Interface Specification
<b>SLI</b>	Structured Light Illuminator (PIXL)
<b>SNS</b>	Amazon Simple Notification Service
<b>SkyCam</b>	Sky Camera (MEDA)
<b>SOAS</b>	Science Operations Analysis Software (or Subsystem)
<b>SOH</b>	State of Health (SuperCam)
<b>SOL</b>	Mars Solar Day
<b>SOWG</b>	Science Operations Working Group
<b>SPaH</b>	Sample Processing and Handling
<b>SPICE</b>	Spacecraft, Planet, Instrument, C-matrix, Events kernels
<b>SQS</b>	Amazon Simple Queue Service
<b>SRAM</b>	Static Random Access Memory
<b>sRGB</b>	Standard RGB (color space)
<b>SuperCam</b>	Super Camera Instrument
<b>SwRI</b>	Southwest Research Institute
<b>TBD</b>	To Be Determined
<b>TBU</b>	Testbed Unit
<b>TDR</b>	Tile-fundamental Data Record
<b>TDS</b>	Telemetry Delivery Subsystem
<b>TIFF</b>	Tagged Image File Format
<b>TRDR</b>	Tile Reduced Data Record
<b>TRN</b>	Translation Relative Navigation (PIXL)
<b>UDR</b>	Unprocessed Data Record
<b>USGS</b>	United States Geological Survey
<b>VCID</b>	Virtual Channel Identifier
<b>VDP</b>	Virtual Data Product
<b>VICAR</b>	Video Image Communication and Retrieval
<b>WATSON</b>	Wide Angle Topographic Sensor for Operations and eNginneering (SHERLOC)
<b>WAV</b>	Audio format (not an acronym)
<b>WEB</b>	Warm Electronic Box
<b>XML</b>	eXtensible Markup Language
<b>XYZ</b>	Cartesian coordinates (not an acronym)

**ZCAM**

Mastcam-Z

## REFERENCES

- [3DTiles] Consortium standard 18-053r2. January 31, 2019.  
<http://www.opengis.net/docs/CS/3DTiles/1.0>.
- [Abarca2019] Abarca, H., Deen R., Hollins G., Zamani P., Maki J., Tinio A., Pariser O., Ayoub F., Toole N., Algermissen S., Soliman T., Lu Y., Golombek M., Calef III F., Grimes K., De Cesare C., Sorice C., “Image and Data Processing for InSight Lander Operations and Science”, Space Science Reviews, Pre-landing InSight Issue, 2019 <https://link.springer.com/article/10.1007/s11214-019-0587-9>
- [Allwood2020] Allwood, Abigail et al., PIXL: Planetary Instrument for X-Ray Lithochemistry, Space Science Reviews volume 216, Article number: 134 (2020)
- [Anderson2010] Anderson, R.C., et al., Mars Science Laboratory Participating Scientists Program Proposal Information Package, December 14, 2010.
- [Bell2012] Bell III, J.F. et al., Mastcam-Z Multispectral Imaging on the Mars Science Laboratory Rover: Wavelength Coverage and Imaging Strategies at the Gale Crater Field Site, *43<sup>rd</sup> Lunar and Planetary Science Conference*, 2012.
- [Bell2017] Bell III, J.F., A. Godber, S. McNair, M.C. Caplinger, J.N. Maki, M.T. Lemmon, J. Van Beek, M.C. Malin, D. Wellington, K.M. Kinch, M.B. Madsen, C. Hardgrove, M.A. Ravine, E. Jensen, D. Harker, R.B. Anderson, K.E. Herkenhoff, R.V. Morris, E. Cisneros, and R.G. Deen, The Mars Science Laboratory Curiosity rover Mast Camera (Mastcam) instruments: Pre-flight and in-flight calibration, validation, and data archiving, *Earth and Space Science*, 4, doi:10.1002/2016EA000219, 2017
- [Bell2020] Bell, III, J.F., J.N. Maki, G.L. Mehall, M.A. Ravine, M.A. Caplinger, Z.J. Bailey, S. Brylow, J.A. Schaffner, K.M. Kinch, M.B. Madsen, A. Winhold, A. Hayes, P. Corlies, M. Barrington, R. Deen, E. Cisneros, E. Jensen, K. Paris, K. Crawford, C. Rojas, L. Mehall, J. Joseph, J.B. Proton, N. Cluff, B. Betts, E. Cloutis, A. Coates, A. Colaprete, K.S. Edgett, B.L. Ehlmann, S. Fagents, J. Grotzinger, C. Tate, C. Hardgrove, K. Herkenhoff, B. Horgan, R. Jaumann, J.R. Johnson, M. Lemmon, G. Paar, M. Caballo-Perucha, S. Gupta, C. Traxler, F. Preusker, M. Rice, M.S. Robinson, N. Schmitz, R. Sullivan, and M.J. Wolff, The Mars 2020 Rover Mast Camera Zoom (Mastcam-Z) Multispectral, Stereoscopic Imaging Investigation, *Space Science Reviews*, in press, <http://dx.doi.org/10.1007/s11214-020-00755-x>, 2020.
- [Bhartia2021] Bhartia Rohit, Luther W. Beegle, Lauren DeFlores, William Abbey, Joseph Razzell Hollis, Kyle Uckert, Brian Monacelli, Kenneth S. Edgett, Megan R. Kennedy, Margarite Sylvia, David Aldrich, Mark Anderson, Sanford A. Asher, Zachary Bailey, Kerry Boyd, Aaron S. Burton, Michael Caffrey, Michael J. Calaway, Robert Calvet, Bruce Cameron, Michael A. Caplinger, Brandi L. Carrier, Nataly Chen, Amy Chen, Matthew J. Clark, Samuel Clegg, Pamela G. Conrad, Moogega Cooper, Kristine N. Davis, Bethany Ehlmann, Linda Facto, Marc D. Fries, Dan H. Garrison, Denine Gasway, F. Tony Ghaemi, Trevor G. Graff, Kevin P. Hand, Cathleen Harris, Jeffrey D. Hein, Nicholas Heinz, Harrison Herzog, Eric Hochberg, Andrew Houck, William F. Hug, Elsa H. Jensen, Linda C. Kah, John Kennedy, Robert Krylo, Johnathan Lam, Mark Lindeman, Justin McGlown, John Michel, Ed Miller, Zachary Mills, Michelle E. Minitti, Fai Mok, James Moore, Kenneth H. Nealson, Anthony Nelson, Raymond Newell, Brian E. Nixon, Daniel A. Nordman, Danielle Nuding, Sonny Orellana, Michael Pauken, Glen Peterson, Randy Pollock, Heather Quinn, Claire Quinto, Michael A. Ravine, Ray Reid, Joe Riendeau, Amy J. Ross, Joshua Sackos, Jacob A. Schaffner, Mark Schwochert, Molly O Shelton, Rufus Simon, Caroline L. Smith, Pablo Sobron, Kimberly Steadman, Andrew

- Steele, Dave Thiessen, Vinh D. Tran, Tony Tsai, Michael Tuite, Eric Tung, Rami Wehbe, Rachel Weinberg, Ryan H. Weiner, Roger C. Wiens, Kenneth Williford, Chris Wollonciej, Yen-Hung Wu, R. Aileen Yingst, Jason Zan  
Perseverance's Scanning Habitable Environments with Raman and Luminescence for Organics and Chemicals (SHERLOC) Investigation  
*Space Science Reviews*, 217, 58  
<https://doi.org/10.1007/s11214-021-00812-z>
- [Bon1991] Bon, B., Algorithm for Using CAHV to Determine SGI Graphics Viewpoint and Perspective, JPL IOM 3472-91-057, August 6, 1991.
- [BPMN] Business Process Model and Notation (BPMN), Object Management Group (OMG) <http://www.omg.org/spec/BPMN>
- [CIE1932] CIE (1932). "Commission internationale de l'eclairage proceedings, 1931." Cambridge University Press Cambridge, 1932.
- [Cozzi] Cozzi, Patrick, Sean Lilley, Gabby Getz (Eds). 3D Tiles Specification 1.0. Open Geospatial
- [CWS] Tinio, A. W., Hollins, G.A., Common Workflow Service: Standards Based Solution for Managing Operational Processes, Third Planetary Data Workshop and the Planetary Geologic Mappers Annual Meeting, 2017
- [Deen1992] Deen, R., "The VICAR File Format", [http://www-mipl.jpl.nasa.gov/external/VICAR\\_file\\_fmt.pdf](http://www-mipl.jpl.nasa.gov/external/VICAR_file_fmt.pdf)
- [Deen2003] Deen, R., Issues with Linearization, JPL Docushare Collection 2700, File 75670, 2003.
- [Deen2005] Deen, R.G. and J.J. Lorre (2005), Seeing in Three Dimensions: Correlation and Triangulation of Mars Exploration Rover Imagery, submitted to 2005 IEEE International Conf. on Systems, Man, and Cybernetics, Waikoloa, Hawaii.
- [NSYT\_Cam\_SIS] Deen, R.G, Payam Zamani, Hallie Abarca, Justin Maki, Jeffrey R. Hall  
InSight Camera EDR and RDR Data Products SIS ID: NSYT-PLD001, JPL D-56937 Release Date: October 1, 2020 [https://pds-imaging.jpl.nasa.gov/data/nsyt/insight\\_cameras/document/insight\\_cameras\\_sis.pdf](https://pds-imaging.jpl.nasa.gov/data/nsyt/insight_cameras/document/insight_cameras_sis.pdf)
- [SIS\_BUNDLE] Deen, R.G. and K. Crombie, "Mars 2020 Project SIS, PDS Camera Archive Bundle Structure", this PDS bundle,  
[urn:nasa:pds:mars2020\\_mission:document\\_camera:mars2020\\_camera\\_bundle\\_sis.pdf](https://pds.nasa.gov/data/nasapds001/mars2020_mission/document/camera/mars2020_camera_bundle_sis.pdf), 2020.
- [Edgett2012] Edgett K. S., R. A. Yingst, M. A. Ravine, M. A. Caplinger, J. N. Maki, F. T. Ghaemi, J. A. Schaffner, J. F. Bell III, L. J. Edwards, K. E. Herkenhoff, E. Heydari, L. C. Kah, M. T. Lemmon, M. E. Minitti, T. S. Olson, T. J. Parker, S. K. Rowland, J. Schieber, R. J. Sullivan, D. Y. Sumner, P. C. Thomas, E. H. Jensen, J. J. Simmonds, A. J. Sengstacken, R. G. Willson, W. Goetz (2012) Curiosity's Mars Hand Lens Imager (MAHLI) investigation, *Space Science Reviews* 170, 259-317  
<https://doi.org/10.1007/s11214-012-9910-4>
- [Edgett2015] Edgett, K. S., M. A. Caplinger, J. N. Maki, M. A. Ravine, F. T. Ghaemi, S. McNair, K. E. Herkenhoff, B. M. Duston, R. G. Willson, R. A. Yingst, M. R. Kennedy, M. E. Minitti, A. J. Sengstacken, K. D. Supulver, L. J. Lipkaman, G. M. Krezoski, M. J. McBride, T. L. Jones, B. E. Nixon, J. K. Van Beek, D. J. Krysak, R. L. Kirk (2015) Curiosity's robotic arm-mounted Mars Hand Lens Imager (MAHLI): Characterization and calibration status, MSL MAHLI Technical Report 0001, version 2. <https://doi.org/10.13140/RG.2.1.3798.5447>
- [Edgett2019] Edgett, K.S., Caplinger, M.A., Ravine, M.A. (2019) Mars 2020 Perseverance SHERLOC WATSON Camera Pre-delivery Characterization and Calibration

Report. Malin Space Science Systems, San Diego, California.  
<https://doi.org/10.13140/RG.2.2.18447.00165>

[FDD\_CECAM] Petrizzo, D., Mars 2020 CEDL Engineering Camera Imaging Functional Design Description (FDD), "Baseline Release, Rev A", JPL D-95849

[FDD\_CEDL] Nuding, D., Mars 2020 CEDL Camera Functional Design Description (FDD), "Baseline Release, Rev A", JPL D-95844

[FDD\_HELI] Petrizzo, D., Mars 2020 Mars Helicopter System Functional Design Description (FDD), "Baseline Release, Rev A", JPL D-99937

[FDD\_MEDA] Diaz, C., Mars 2020 CEDL and Surface MEDA Functional Design Description (FDD), "Baseline Release, Rev A", JPL D-95851

[FDD\_PIXL] Ervin, J., Mars 2020 CEDL and Surface PIXL Functional Design Description (FDD), "Baseline Release, Rev A", JPL D-95857

[FDD\_PPPCS] Alibay, Farah, M2020 Pointing, Positioning, Phasing & Coordinate Systems Introduction and Coordinate Systems, JPL D-34641

[FDD\_RAPP] Alibay, Farah Mars 2020 Rover Attitude Positioning and Pointing FDD, JPL D-95865

[FDD\_SAPP] M2020 Pointing, Positioning, Phasing & Coordinate Systems Introduction and Coordinate Systems, Farah Alibay, JPL D-34641

[FDD\_SCAM] Gontijo, I., Mars 2020 CEDL and Surface SuperCam Functional Design Description (FDD), "Baseline Release, Rev A", JPL D-95868

[FDD\_SECAM] Petrizzo, D., Mars 2020 Surface Engineering Camera Imaging Functional Design Description (FDD), "Baseline Release, Rev A", JPL D-96164

[FDD\_SHERLOC] Bailey, Z., Mars 2020 CEDL Surface SHERLOC Functional Design Description (FDD), "Baseline Release, Rev A", JPL D-95867

[FDD\_ZCAM] Bailey, Z., Mars 2020 CEDL and Surface Mastcam-Z Functional Design Description (FDD), "Baseline Release, Rev A", JPL D-95850

[Fuhrmann2014] Fuhrmann, Simon and Michael Goesele. Floating Scale Surface Reconstruction. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2014), Vancouver, Canada, 2014.

[Gennery1986] Gennery, D., Camera Calibration, JPL IOM 347/86/10, February 5, 1986.

[Gennery1987] Gennery, D. et al., Sensing and Perception Research for Space Telerobotics at JPL, *Proceedings of the IEEE Intern. Conf. on Robotics and Automation*, March 31 - April 3, 1987.

[Gennery1991] Gennery, D., Camera Calibration Including Lens Distortion, JPL D-8580, May 31, 1991.

[Gennery1994] Gennery, D., Inclusion of Old Internal Camera Model in New Calibration, JPL IOM 386.3-94-001, February 22, 1994.

[Gennery2001a] Gennery, D., "Least-Squares Camera Calibration Including Lens Distortion and Automatic Editing of Calibration Points", Calibration and Orientation of Cameras in Computer Vision, ISBN 3-540-65283-3, 2001.

[Gennery2001b] Gennery, D., Computations for Generalized Camera Model Including Entrance, Part 1 and Part 2, unpublished, May 23, 2001.

[Gennery2002] Gennery, D., Generalized Camera Calibration Including Fish-Eye Lenses, JPL D-03-0869, 2002.

[GeoJSON] GeoJSON Format (Internet Engineering Task Force (IETF) RFC 7946: <https://tools.ietf.org/html/rfc7946>)

[Hayes2020] Hayes, A.G., P. Corlies, C. Tate, J.F. Bell III, J.N. Maki, M. Caplinger, K.M. Kinch, K. Herkenhoff, B. Horgan, J. Johnson, G. Parr, M.S. Rice, E. Jensen, T.M. Kubacki, E. Cloutis, B. Ehlmann, E. Lakdawalla, R. Sullivan, A. Winhold, M. Barrington, A. Parkinson, J. van Beek, P. Caballo-Perucha, E. Cisneros, D. Dixon, C. Donaldson, O.B. Jensen, J. Kuik, K. Lapo, and A. Magee, Pre-Flight

- Calibration of the Mars 2020 Rover Mastcam Zoom (Mastcam-Z) Multispectral, Stereoscopic Imager, submitted to *Space Sci. Rev.*, 2020.
- [ITU2015] ITU (2015). International Telecommunication Union Radiocommunication Sector, Recommendation ITU-R BT.709-6, "Parameter values for the HDTV standards for production and international programme exchange", June 2015, BT Series, Broadcasting Service, Television, Electronic Publication Geneva, 2015. <http://www.itu.int/rec/R-REC-BT.709-6-201506-I/en>
- [Kazhdan2006] Kazhdan. M., Matthew Bolitho, Hugues Hoppe. Poisson surface reconstruction. Symposium on Geometry Processing 2006, 61-70.
- [Kazhdan2010] Kazhdan, Surendran, Hoppe. Distributed Gradient-Domain Processing of Planar and Spherical Images. ACM Transactions On Graphics, Vol 29, No 2, April 2010.
- [Kiely2004] Kiely, Aaron, Mars Exploration Rover (MER) Project ICER User's Guide, MER 420-8-0538, JPL D-22103, January 5, 2004.
- [Kinch2020] Kinch, K.M., M.B. Madsen, J.F. Bell III, J.N. Maki, Z. Bailey, A.G. Hayes, O.B. Jensen, M. Merusi, M.H. Bernth, A.N. Sørensen, M. Hilverda, E. Cloutis, D. Applin, E. Mateo-Marti, J. A. Manrique, G. Lopez-Reyes, A. Bello-Arufe, B. Ehlmann, J. Buz, A. Pommerol, N. Thomas, L. Affolter, K. Herkenhoff, J.R. Johnson, M. Rice, P. Corlies, C. Tate, M. Caplinger, E. Jensen, T. Kubacki, E. Cisneros, K. Paris, A. Winhold, and D. Wellington, Radiometric Calibration Targets for the Mastcam-Z Camera on the Mars 2020 Rover Mission, *Space Sci. Rev.*, 216, 141, [doi:10.1007/s11214-020-00774-8](https://doi.org/10.1007/s11214-020-00774-8), 2020.
- [Kraus2007] Karl Kraus. Photogrammetry. Geometry from Images and Laser Scans. Second Edition. Part 1. De Gruyter.2007 <https://doi.org/10.1515/9783110892871>
- [Litwin2020] Litwin, Todd E., PSPH: The Planospheric Camera Model, 2020 (Unpublished). <https://github.jpl.nasa.gov/pages/telitwin/ccal/doc/www/planospheric.pdf>
- [Maki2018] Maki, J. N., Golombek M., Deen R., Abarca H., Sorice C., Goodsall T., Schwochert M., Lemmon M., Trebi-Ollennu A., Bannert W.B., "The Color Cameras on the InSight Lander", Space Science Reviews, Pre-landing InSight Issue, 2018, <https://link.springer.com/article/10.1007%2Fs11214-018-0536-z>.
- [Maki2020] Maki, J.N., et al., The Mars 2020 Engineering Cameras and Microphone on the Perseverance Rover: A Next-Generation Imaging System for Mars Exploration, Manuscript Draft for Space Science Reviews, 2020.
- [Malvar2004] Malvar, H.S., Li-Wei He, and R. Cutler, "High-quality linear interpolation for demosaicing of Bayer-patterned color images", Proceedings, IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), doi: 10.1109/ICASSP.2004.1326587, 2004.
- [OBJ] Object Files (.obj) <http://paulbourke.net/dataformats/obj/>
- [Olson2007] Olson Clark F., Larry H. Matthies, John R. Wright, Rongxing Li, Kaichang Di, "Visual terrain mapping for Mars exploration" Computer Vision and Image Understanding Vol 105(1), pp 73-85, 2007.
- [OpenInventor] Open Inventor Toolkit. <https://openinventor.com/>
- [OpenCV] OpenCV. Open Source Computer Vision library of programming functions. [https://docs.opencv.org/master/dc/dbb/tutorial\\_py\\_calibration.html](https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html)
- [PDSConcept] PDS4 Concepts, version 1.14.0, May 19, 2020, [https://pds.nasa.gov/datastandards/documents/concepts/Concepts\\_1.14.0.pdf](https://pds.nasa.gov/datastandards/documents/concepts/Concepts_1.14.0.pdf)
- [PDSDPH] The PDS4 Data Provider's Handbook, version 1.14.0, May 19, 2020, [https://pds.nasa.gov/datastandards/documents/dph/current/PDS4\\_DataProvidersHandbook\\_1.14.0.pdf](https://pds.nasa.gov/datastandards/documents/dph/current/PDS4_DataProvidersHandbook_1.14.0.pdf)
- [PDSIM] PDS4 Information Model Specification, version 1.14.0, Mar 23, 2020, [https://pds.nasa.gov/datastandards/documents/im/current/index\\_1E00.html](https://pds.nasa.gov/datastandards/documents/im/current/index_1E00.html)



- [PDSSStd] Planetary Data System Standards Reference, version 1.14.0, May 22, 2020, [https://pds.nasa.gov/datastandards/documents/sr/current/StdRef\\_1.14.0.pdf](https://pds.nasa.gov/datastandards/documents/sr/current/StdRef_1.14.0.pdf)
- [Serricchio2021] Serricchio, F., San Martin, A.M., Brugarolas, P., and Casoliva, J., "The Mars 2020 Perseverance Navigation Filter," Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, AIAA, Big Sky, MT, Aug. 2021.
- [Shah2017] Shah, B., Mars 2020 Flight-Ground Interface Control Document (FGICD), "Volume 1, Downlink, Rev A, Version 1.0", JPL D-95521, October 3, 2017
- [SIS\_MEDA] M2020 MEDA EDR SIS (Draft)
- [SIS\_MOXIE] M2020 MOXIE EDR SIS (Draft)
- [SIS\_PIXL] M2020 PIXL EDR Non-imaging SIS (Draft)
- [SIS\_RIMFAX] M2020 RIMFAX EDR SIS (Draft)
- [SIS\_SCAM] M2020 SuperCam EDR SIS (Draft)
- [SIS\_SHERLOC] M2020 SHERLOC Non-Imaging SIS (Draft)
- [Wright2005] John R. Wright, Ashitey Trebi-Ollennu, Frank Hartman, Brian Cooper, Scott Maxwell, Jen Yen, Jack Morrison, "Terrain Modelling for In-situ Activity Planning and Rehearsal for the Mars Exploration Rovers" IEEE Int. Conf. on Systems, Man and Cybernetics, Vol 2, pp 1372- 1377, 2005.
- [Wyszecki1982] Wyszecki, G., Stiles, W.S. (1982). "Color Science: Concepts and Methods, and Formulae", 2nd ed., John Wiley, New York, 1982.
- [Yakimovsky1977] Yakimovsky, Y. and Cunningham R., A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras, January 7, 1977.

# 1. INTRODUCTION

## 1.1 Purpose and Scope

The purpose of this Data Product Software Interface Specification (SIS) is to provide consumers of M2020 camera data products with a detailed description of the products and how they are generated. Content in this document supports data products generated by the Instrument Data Subsystem (IDS) as well as the camera teams for the following instruments:

- Engineering Camera (ECAM) suite:
  - a. Navigation Cameras (Navcams)
  - b. Hazard Avoidance Cameras (Hazcams)
  - c. Sample Caching System Camera (CacheCam)
- SuperCam Remote Micro-Imager (RMI)
- Mastcam-Z (ZCAM) instrument suite:
  - a. Mastcam-Z Left (ZCAM-L)
  - b. Mastcam-Z Right (ZCAM-R)
- SHERLOC camera suite
  - a. SHERLOC-WATSON
  - b. SHERLOC-ACI
- PIXL Micro Context Camera (MCC)
- MEDA Skycam
- EDL camera suite
  - a. Parachute Uplook Cameras (PUC)
  - b. Descent Stage Downlook Camera (DDC)
  - c. Rover Uplook Camera (RUC)
  - d. Rover Downlook Camera (RDC)
  - e. EDL Microphone
- Lander Vision System Camera (LCAM)
- Helicopter camera suite
  - a. Helicopter Navigation Camera
  - b. Helicopter Return To Earth Camera

This document covers both operations and archive products for Mars 2020 camera data. The users for whom this SIS is intended thus include the Mars 2020 operations team, the Mars 2020 science team, and the general public via the PDS Archive.

## 1.2 How To Use This Document

This SIS is a standalone reference document for camera data across all instruments. It is intended to describe the products in sufficient detail for them to be used for scientific investigations long after the mission is over, as well as supporting Mars 2020 operations. However, in any such complex system as the Mars 2020 rover, it is necessarily incomplete in some areas. The reader is encouraged to use the references to fill in any gaps or details that may exist.

Four of the instruments covered here (SHERLOC, PIXL, SuperCam, and MEDA) are not primarily camera instruments, but include cameras as a necessary part of their operations. For these, only aspects relating to the cameras are described, not information about the instrument in general. Again, the reader is encouraged to refer to the PDS bundles for each of these instruments, which contain documentation specific to that instrument, covering the entire instrument beyond just the cameras.

Great care has been taken to make data from all the cameras across all instruments as similar as possible. This facilitates using common software to process all camera data, regardless of instrument. While there are sections describing camera-specific elements, most of this document applies to data from any camera. There are obvious exceptions which are not explicitly stated; for example, it is physically impossible to take stereo data with the MEDA-Skycam and CacheCam instruments. Those sections which do not apply to a given camera should be ignored. However, sometimes camera capabilities can be surprising: it is entirely possible to do stereo from the arm-mounted cameras by moving the arm, for example. While this is not done systematically, it may be done via special processing and the products so produced match the specifications in this document.

Metadata (aka labels) are a critical feature of Mars 2020 camera data. All camera products have an extremely rich set of labels, which describe conditions under which the image was taken as well as processing that has occurred to the image. Label items (for both PDS4 and VICAR/ODL) are documented in various forms in Appendix B (which consists of several tables that are physically separate from this document, but logically part of it). These tables are the primary reference for label items and are a large part of this document. Many labels are discussed throughout the text as needed, but the Appendix B tables should be the first place the reader goes to learn about label items.

Important for understanding image files is the three-letter Product Type Code that is used to refer to product types. This code is used throughout the document, and the codes are listed in Section 18.

The text of this SIS can be organized into several broad categories.

- Section 2 provides an overview of the cameras themselves
- Sections 3 and 4 describe camera images in general terms
- Sections 5 and 6 describe the EDR and FDR products, which are the foundational products
- Sections 7 through 14 describe the derived products (RDRs) that result from image processing
- Sections 15 and 16 discuss at a high level how the products are made and tangentially related topics
- Sections 18 through 20.11 discuss the type codes, filenames and coordinate systems
- Appendix A lists the telemetry products from which the images come
- Appendix B describes the all-important labels
- Appendix C describes the calibration data

This document has been reviewed and determined not to contain export controlled technical data.

- Appendix D describes the pipeline data flow
- Appendix E contains supplemental information for SHERLOC Watson and ACI images
- Appendix F summarizes the various SCLKs in filenames and labels
- Appendix G summarizes the Special Product Flags in use
- Appendix H contains summary tables of camera characteristics

Note that many of the appendices contain tables or diagrams that are stored in separate physical files in the archive, with separate labels, as noted in the appendices. These files are considered logically part of the SIS, along with this main document.

## 1.3 Constraints and Applicable Documents

This SIS is meant to be consistent with the contract negotiated between the M2020 Project and the M2020 Principal Investigators (PI) for the Engineering cameras, the SuperCam RMI, the Mastcam-Z cameras, the Sherlock-Imaging (WATSON) and ACI cameras, the PIXL Micro Context Camera, and the MEDA Sky Camera, in which reduced data records and documentation are explicitly defined as deliverable products. Because this SIS governs the specification of data products used during mission operations, any proposed changes to this SIS must be impacted by all affected software subsystems observing this SIS in support of operations (e.g., RPS, IDS, SOAS).

Product label keywords may be added to future revisions of this SIS. Therefore, it is recommended that software designed to process EDRs and RDRs specified by this SIS should be robust to (new) unrecognized keywords. Similarly, entirely new products may be added over time.

This Data Product SIS is responsive to the following M2020 documents listed in References section.

## 2. INSTRUMENT OVERVIEW

In this section, overviews are provided for the all of M2020 camera instruments mentioned in Section 1.1. The M2020 rover instrument payload includes 25 individual cameras (see Figure 2-2). The main differences between these instruments are in the optics, mounted position, and articulation methods. The cameras are color, except for the SHERLOC-ACI, PIXL MCC, MEDA SkyCam, and Helicopter Navigation camera. Of the 25 cameras, there are 5 sets of stereo pairs and 15 single cameras, as listed in Table 2-1 below. A more complete summary of camera characteristics can be found in Appendix H.

**Table 2-1 – Tabulation of M2020 Camera Instruments**

<b>M2020 Camera Instrument</b>	<b>Location</b>	<b>Number</b>	<b>Color</b>	<b>Bits</b>	<b>Resolution (SxL)</b>
Navcam	Stereo pair on RSM	2	Yes	12	5120 x 3840
Front Hazcam	Stereo pair at front of WEB	4	Yes	12	5120 x 3840
Rear Hazcam	Stereo pair at rear of WEB	2	Yes	12	5120 x 3840
CacheCam	Monoscopic inside WEB	1	Yes	12	5120 x 3840
SuperCam RMI	Monoscopic on RSM	1	Yes	10-13	2048 x 2048
Mastcam-Z	Stereo pair on RSM	2	Yes	11	1648 x 1200
SHERLOC-WATSON	Monoscopic on robotic arm turret	1	Yes	11	1648 x 1200
SHERLOC-ACI	Monoscopic on robotic arm turret	1	No	11	1648 x 1200
MEDA SkyCam	Monoscopic on rover deck	1	No	8	1024 x 1024
PIXL MCC	Monoscopic on robotic arm turret	1	No	8	752 x 580
EDL PUC	Monoscopic on top of backshell	3	Yes	8	1280 x 1024
EDL DDC	Monoscopic on bottom of descent stage structure	1	Yes	8	1280 x 1024
EDL RUC	Monoscopic on top of rover deck	1	Yes	8	1280 x 1024
EDL RDC	Monoscopic on side of WEB	1	Yes	8	1280 x 1024
LCAM	Monoscopic on the front left corner of rover body	1	No	8	1024 x 1024
Helicopter Nav Cam	Downward- looking on Helicopter Body	1	No	8	640 x 480
Helicopter RTE Cam	Side-looking on Helicopter Body	1	Yes	8	4224 x 3120
	<b>Total</b>	<b>25</b>			

There are several color capable cameras onboard the rover and they all utilize a Bayer filter/pattern, consisting of repeated squares of red (R), green (G), and blue (B) filters, as shown in Figure 2-1.

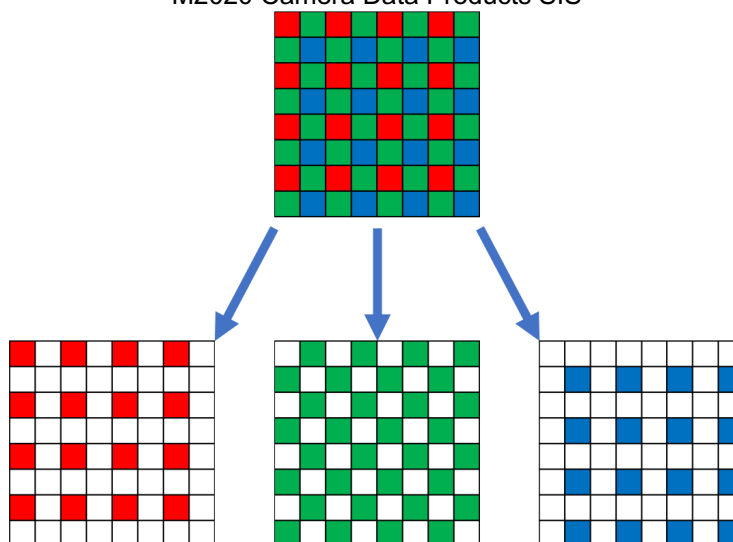


Figure 2-1: Example Bayer filter pattern

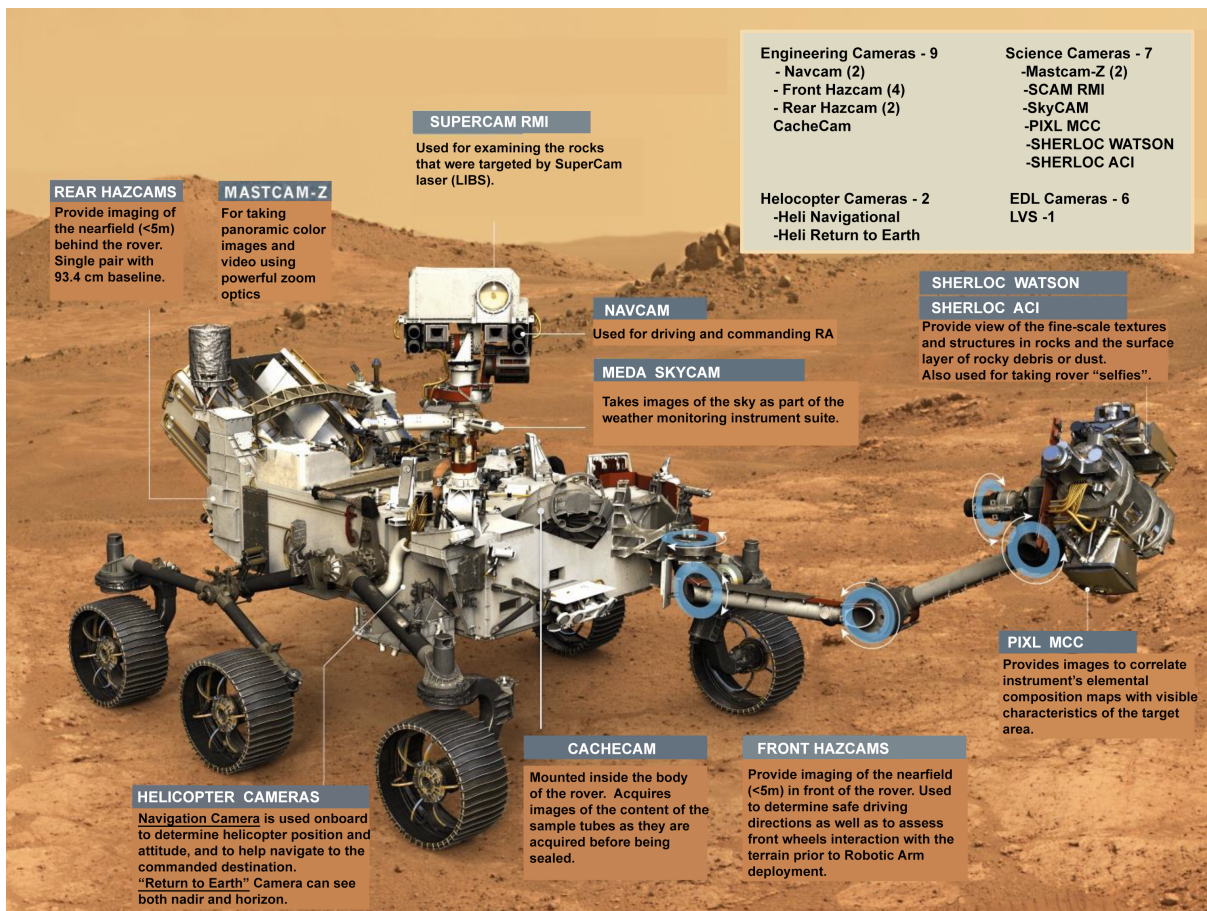
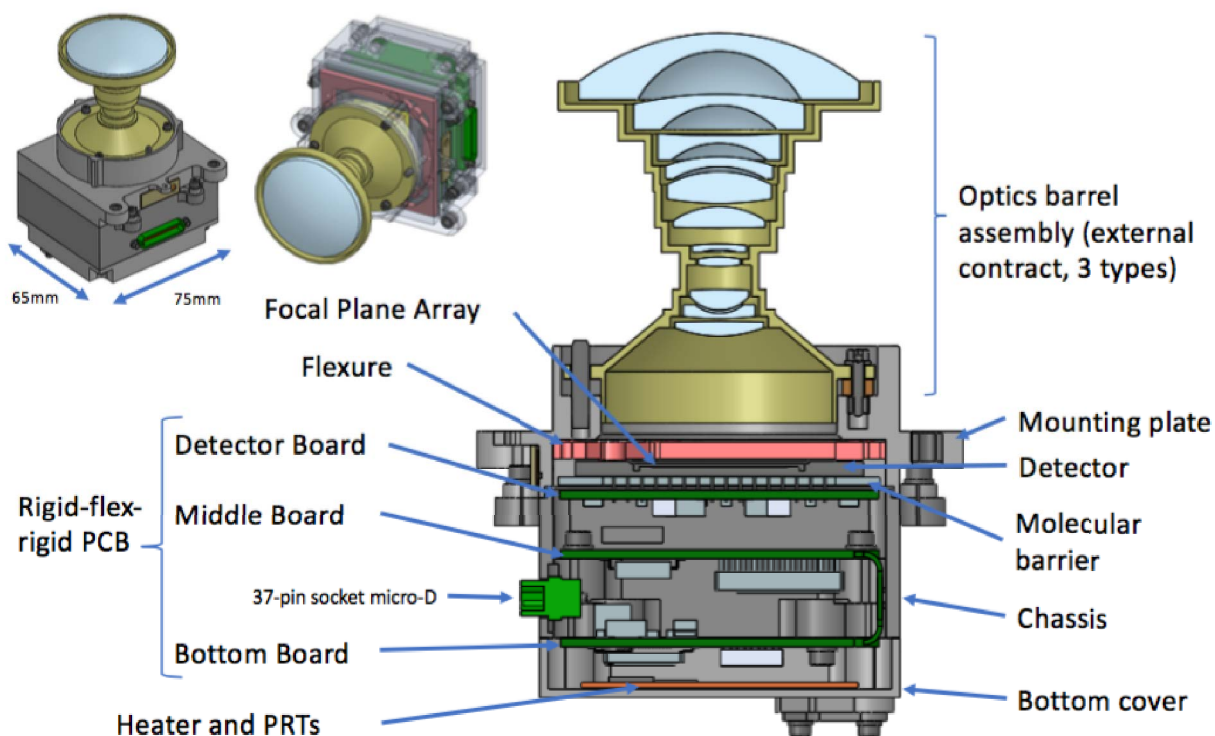


Figure 2-2: The cameras on M2020 Rover

## 2.1 Engineering Camera Instrument Suite

Each Engineering Camera (ECAM) is composed of a detector and electronics within a single mechanical housing, specifically: 1) an optical lens assembly, 2) a CMOS image sensor and 3) electronics that includes, a Field Programmable Gate Array, memory, and a legacy interface electronics to the rover. All of the ECAMs are based on a new Enhanced Engineering Camera (EECAM) design, which uses 5120 x 3840, 20 megapixel global shutter CMOS image sensors with a Bayer filter/pattern as shown in Figure 2-3:. Because of the electronics commonality, image data from all engineering cameras are functionally equivalent. The cameras have an external film heater that warms up the electronics to above the minimum operating temperature of -55 deg. C. The rover provides supply voltage of +5.5 V to the cameras. Each camera weighs less than 750 grams and uses approximately 4.5 Watts of power.



**Figure 2-3: Engineering Camera hardware design**

### 2.1.1 Hazard Avoidance Camera (Hazcam)

The Hazard Avoidance Cameras (Hazcams) are two stereo pairs (front) and one stereo pair (rear) of engineering cameras with fish-eye lenses at 25 cm (front) and 93.4 cm (rear) baseline separation mounted at both the front and rear ends of the Warm Electronics Box (WEB).

The Hazcams provide imaging primarily of the near field (< 5 m) both in front of and behind the rover. These cameras will be used to determine safe driving directions for the rover and provide for on-board hazard detection using stereo data to build range maps. They also support science operations for selecting near field target and robotic arm operations.

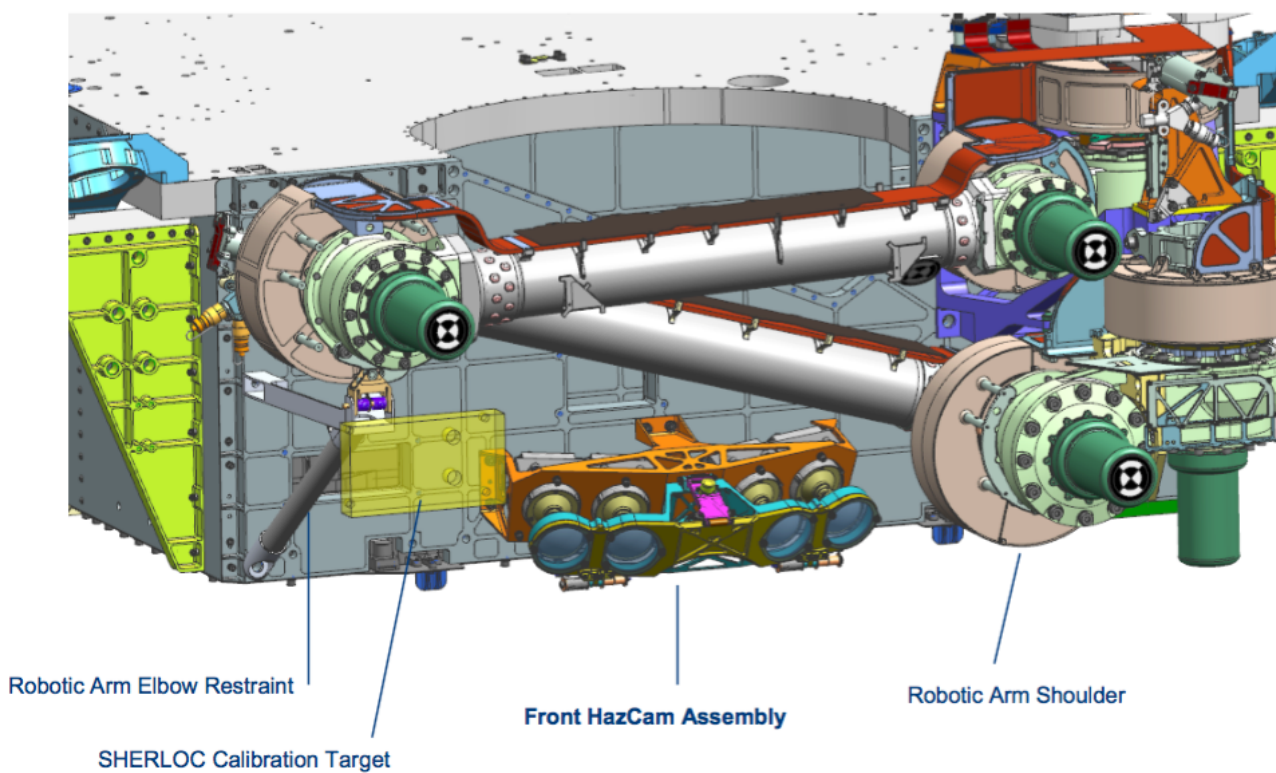
Hazcam optics characteristics useful in the analysis of EDR and RDR products are described in Table 2-2. Front Hazcam locations are shown in Figure 2-4 with the mapping to names in Figure 2-5. Rear Hazcam locations are shown in Figure 2-6.

Note that there are two sets of front Hazcams, one connected to each Rover Compute Element (RCE). If RCE-A is active, Front Haz A cameras will be used, while the B cameras are used with RCE-B. There is just one pair of rear Hazcams, with an unusually large baseline since one is on each side of the rover body.

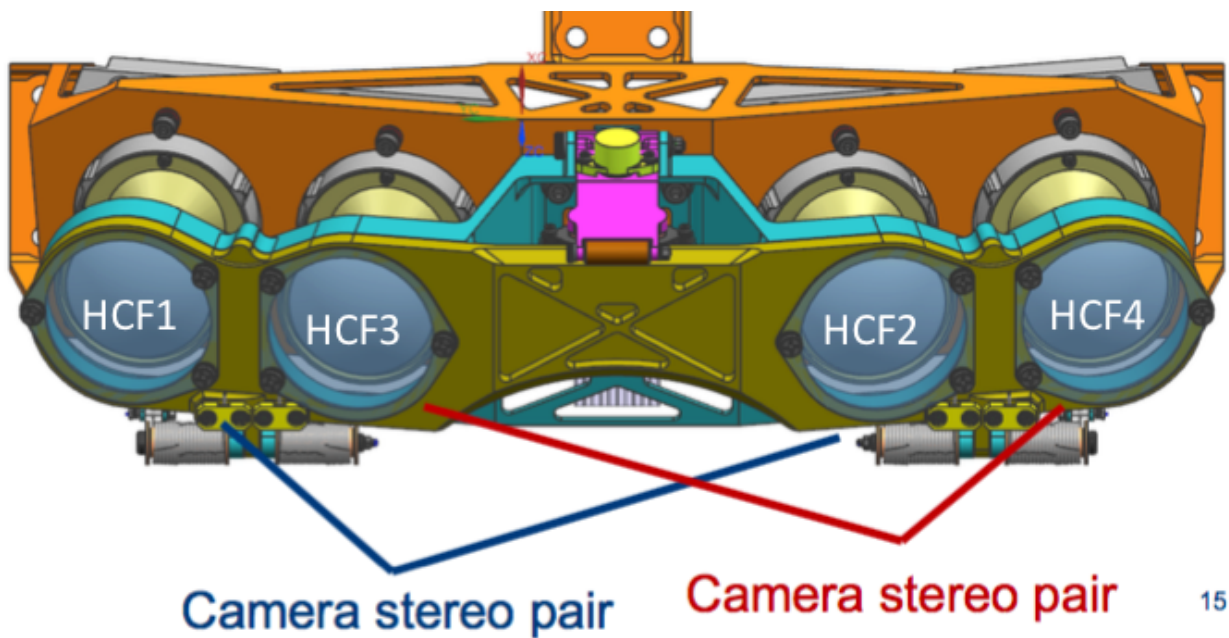
**Table 2-2 – Hazcam operational characteristics**

Characteristic	Value
Resolution (S x L)	5120 x 3840
Bit Depth	12
Field of View (FOV)	136 x 102 deg
Baseline Stereo Separation	24.8 cm for front, 93.4 cm for rear
Angular Resolution	0.46 mrad/pixel at center
Pixel Size	6.4 $\mu\text{m}$ x 6.4 $\mu\text{m}$
Spectral Bandpass	400 - 675 nm
Focal Length	13.8 mm
f/number	12
Depth of Field	0.8 m - infinity
Best Focus	0.9 m

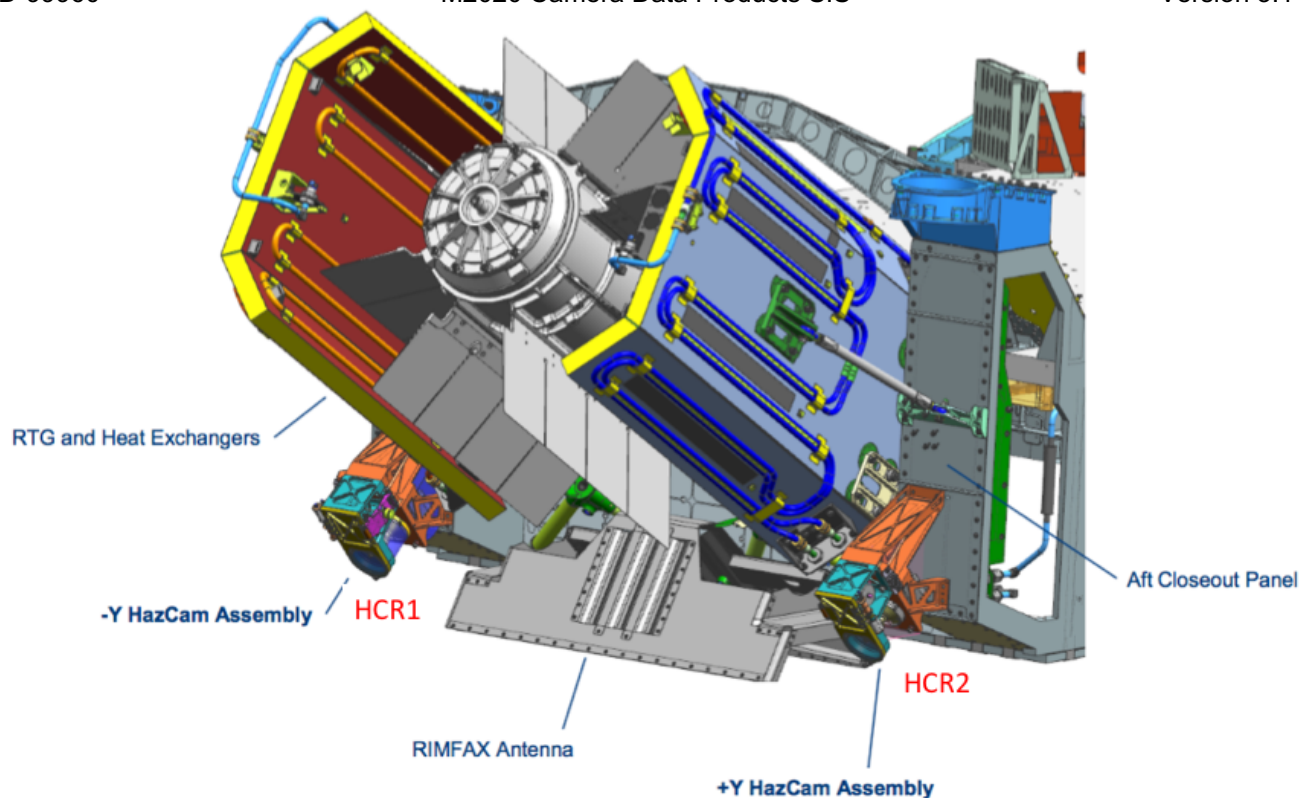




**Figure 2-4: Front Hazcam locations**



**Figure 2-5: Front Hazcam stereo pair mapping. Blue is the A-side pair, red is the B-side pair.**



**Figure 2-6: Rear Hazcam location**

### 2.1.2 Navigation Camera (Navcam)

The Navigation Camera (Navcam) is a mast-mounted stereo pair of engineering cameras at 42.4 cm baseline separation with a spectral bandpass at approximately 275 nm (passing longer wavelengths). It will primarily be used for navigation purposes and general site characterization (360° panoramic images and targeted images of interest, including terrain not viewable by the Hazcams).

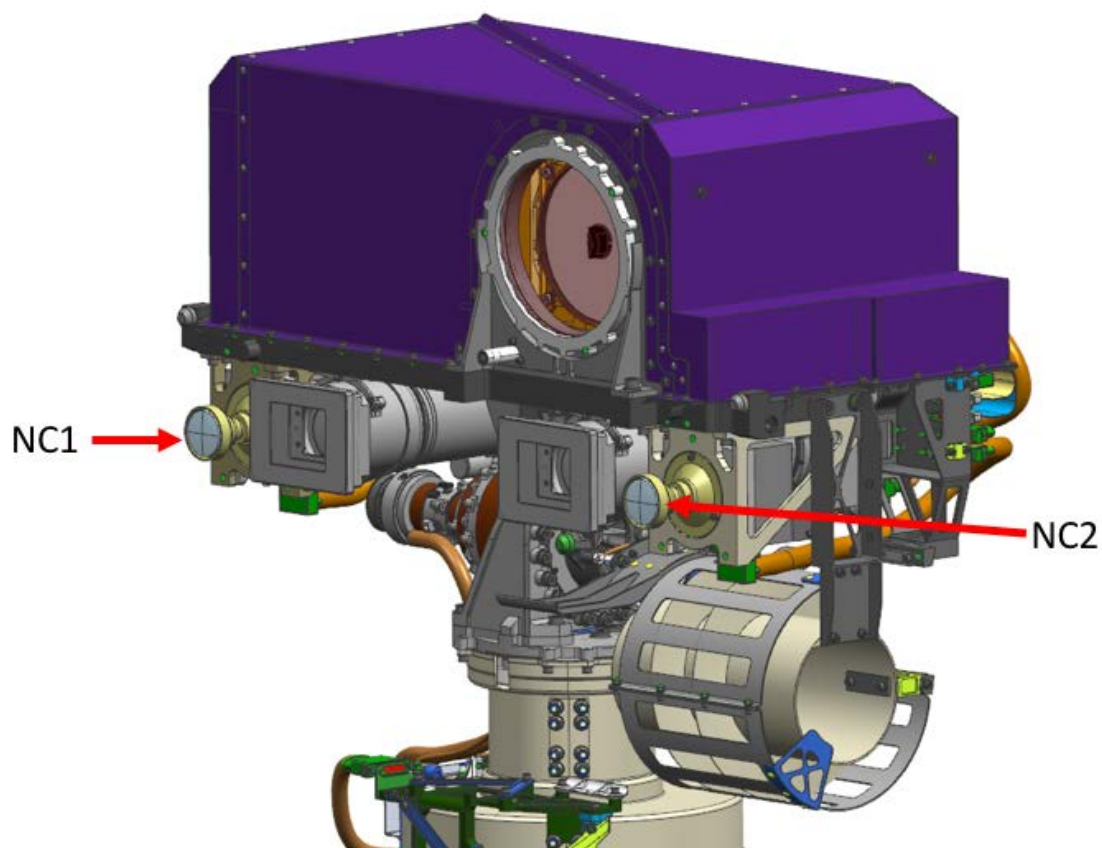
The cameras are co-boresighted with the Mastcam-Z, and Navcam images will also be used for Science target selection and analysis.

Navcam optics characteristics useful in the analysis of EDR and RDR products are described in Table 2-3, with Navcam location shown in Figure 2-7:

**Table 2-3 – Navcam Operational Characteristics**

Characteristic	Value
Resolution (S x L)	5120 x 3840
Depth Bits	12
Field of View (FOV)	96° x 73°
Baseline Stereo Separation	42.4 cm
Angular Resolution	0.33 mrad/pixel at center

Characteristic	Value
Pixel Size	6.4 $\mu\text{m}$ x 6.4 $\mu\text{m}$
Spectral Bandpass	400 – 675 nm
Focal Length	19.1 mm
f/number	12
Depth of Field	1.5 m - infinity
Best Focus	3.5 m



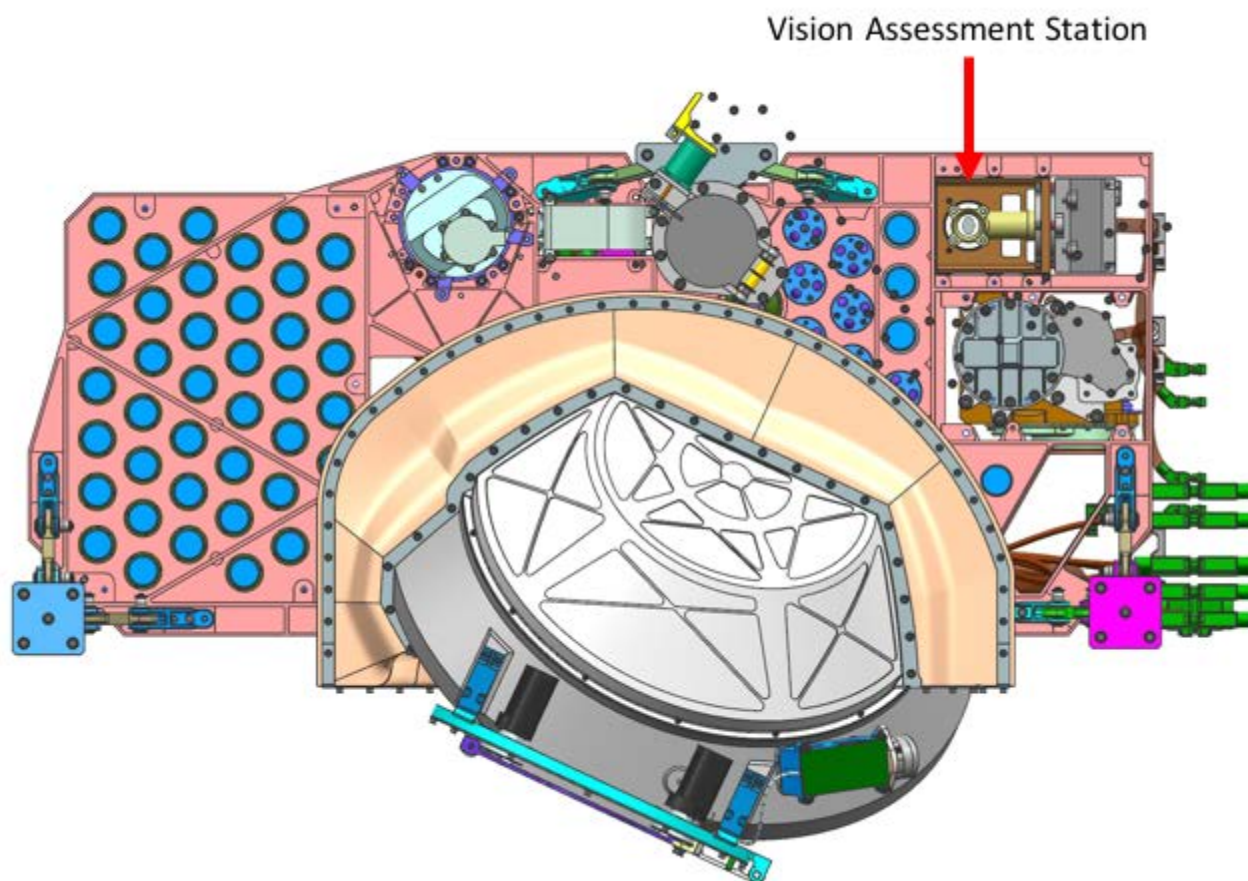
**Figure 2-7: Navcam location**

### 2.1.3 Sample Caching System Camera (CacheCam)

The CacheCam is a single camera mounted inside the body of the rover. It acquires images of the contents of the sample tubes as they are acquired before being sealed. It does not articulate, although the tube carrier can articulate. Note that the CacheCam does not have a camera model associated with it. Table 2-4 provides the camera characteristics, while Figure 2-8 shows the location of the camera.

**Table 2-4 – CacheCam Operational Characteristics**

Characteristic	Value
Resolution (S x L)	5120 x 3840
Bit Depth	12
Field of View (FOV) Horizontal	65 mm
Field of View (FOV) Vertical	49 mm
Field of View (FOV) Diagonal	81 mm
Angular Resolution	12.5 um/pix at plane of focus ~0.51x magnification
Pixel scale (center of FOV)	12.5 microns/pixel
Focal Length	32 mm
Focal ratio	f/8
Depth of Field	+ / - 5 mm
Best Focus	140 mm below illuminator mirror

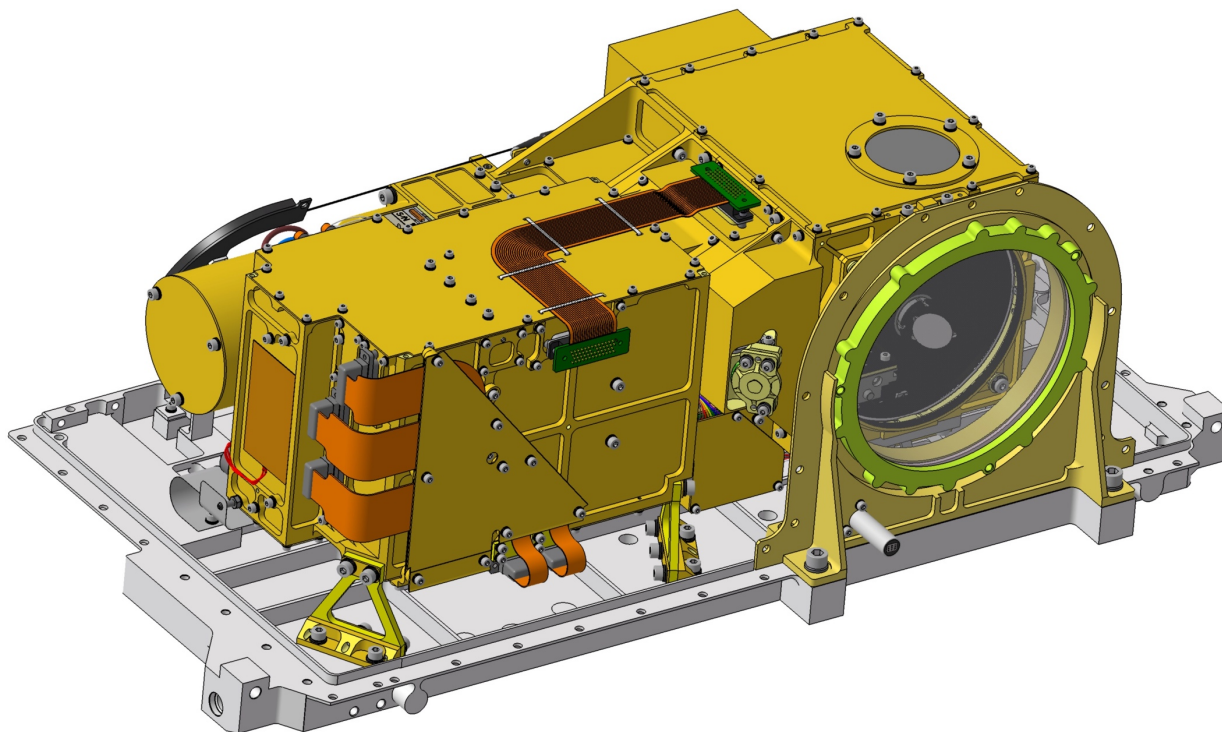
**Figure 2-8: CacheCam location (yellow cylinder in Vision Assessment Station)**

This document has been reviewed and determined not to contain export controlled technical data.



## 2.2 SuperCam Remote Micro-Imager (RMI)

The SuperCam RMI camera subsystem is packaged onboard the rover in two instrumentation units: the Mast Unit affixed to the RSM (Figure 2-9), and the Body Unit housed within the rover WEB. The RMI camera provides high-resolution color context images of the LIBS sampling area as well as imaging rock morphologies and distant features. It includes an adjustable focus capability with 2048 x 2048 CMOS color detector (Bayer Filter). Raw RMI images are 10-to13-bit (coded on 16-bits) and are often transformed onboard into four images of 1024 x 1024 pixels each consisting of the red, green1, green2, and blue (RGGGB) Bayer cells in separate images [FDD\_SCAM].



**Figure 2-9: SuperCam Mast Unit**

RMI optics characteristics useful in the analysis of EDR and RDR products are described in Table 2-5 below. RMI image data return compression modes are listed in Table 2-6. The camera acquisition of the scene and subsequent onboard storage and readout of image data is illustrated in Figure 2-10.

**Table 2-5 – SCAM RMI Operational Characteristics**

Characteristic	Value
----------------	-------

This document has been reviewed and determined not to contain export controlled technical data.

Characteristic	Value
Resolution (S x L)	2048 x 2048
Bit Depth	12 (13 for HDR)
Field of View (FOV) useful	18.8 mrad
Field of View (FOV) full frame, including dark corners	20.6 mrad
Spatial Resolution with 20% contrast	80 $\mu$ rad/pixel
Angular Resolution	10 $\mu$ rad/pixel
Spectral Bandpass	375 - 655 nm
Focus Range	~1.05 m to infinity
Depth of Field	~2.5 mm at 1.56m ~9.5mm at 3m ~10.5 cm at 10m

Table 2-6 – RMI Compression Modes

Mode	Description
0	No compression
1	ICER
2	JPEG

## RMI Image Data Formats

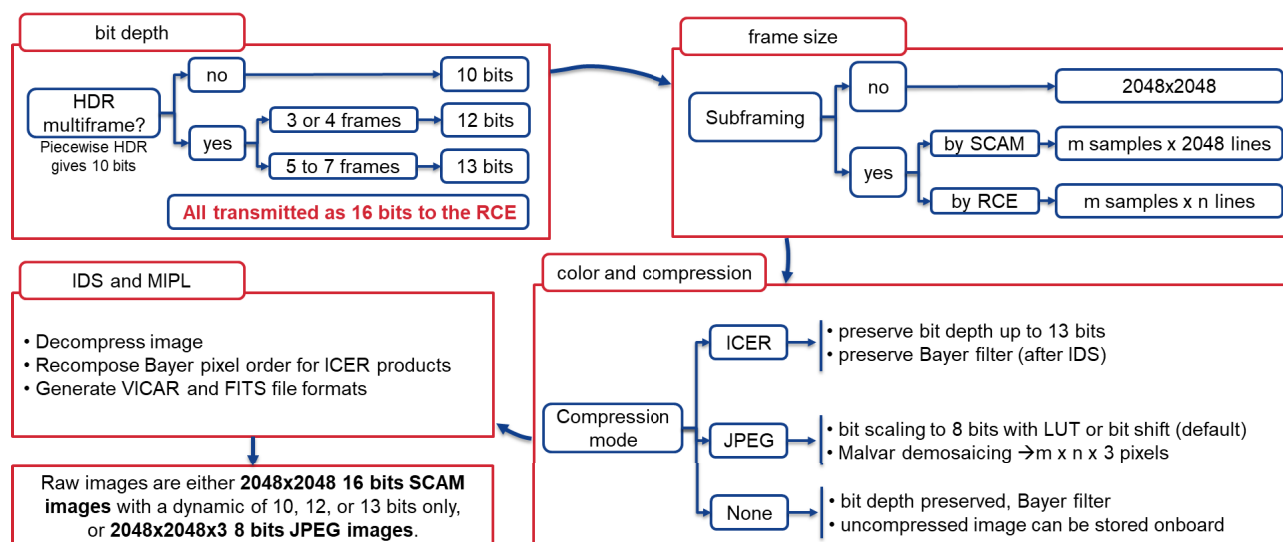


Figure 2-10: RMI Image Data Formats

## 2.3 Mastcam-Z Camera Instrument Suite

Mastcam-Z is comprised of a pair of color-capable zoomable, focusable stereo cameras (“eyes”) mounted on the rover’s Remote Sensing Mast (RSM). Each camera has identical zoom/focus capabilities and a different set of spectral filters. The CCDs consists of 1648 x 1200 pixels, where 1600 x 1200 are photoactive pixels. These cameras are capable of video. They acquire color via Bayer-pattern filters on the CCD, but also have selectable filters for science/geology applications. For more details on how the Bayer pattern interacts with filters, see Section 5.6.3.

Mastcam-Z optics characteristics that are useful in the analysis of EDR and RDR products are described in Table 2-7.

**Table 2-7 – Mastcam-Z Operational Characteristics**

Characteristic	Left/Right Eye
Resolution (S x L)	1648 x 1200 (1600 x 1200 photoactive pixels)
Bit Depth	11
Field of View (FOV)	Wide: 25.6° x 19.1° Narrow: 6.18° x 4.63°
Baseline Stereo Separation	24.4 cm
Toe-in per eye	1.15°
Pixel Size	7.4 $\mu\text{m}$ x 7.4 $\mu\text{m}$
Spatial Resolution (Wide Zoom)	~0.6 mm/pix @ ~2 m range ~3 cm/pix @ 100 m range
Spatial Resolution (Narrow Zoom)	~0.13 mm/pix @ ~2 m range ~6.7 mm/pix @ 100 m range
Angular Resolution	26 mm focal length: 283 $\mu\text{rad/pix}$ 110 mm focal length: 67.4 $\mu\text{rad/pix}$
Focal Length	26 – 110 mm
f/number	26 mm focal length: f/6.7 110 mm focal length: f/9.5
Focus Range	0.5m to $\infty$ for 26 to ~50mm 1m to $\infty$ for ~50 to 110mm
Number of Spectral Filters	7 plus Broadband RGB Bayer pattern

Each Mastcam-Z camera has an 8-position filter wheel (Table 2-8). One of the positions per camera is a broadband filter for use with the RGB Bayer color capability of the CCD. Six of the remaining positions per camera provide color imaging capability in eleven unique wavelengths from 400 to 1000 nm. One additional filter per camera has a neutral density coating to provide direct solar imaging capability in RGB (left eye) and 880 nm (right eye). The remaining six narrow-band filters are used for science multispectral imaging.

Mastcam-Z multispectral, stereo, and panoramic images will use the RGB and narrow-band filters to provide detailed geologic and geographic information. These images will explore the morphology,

topography, and geologic context along the rover's traverse in addition to constraining mineralogic, photometric, and physical properties of the surface. The images will target places exhibiting color variations, diagnostic morphologies, and potentially diverse chemistries and lithologies. These images will monitor and characterize atmospheric and astronomical occurrences. In addition, Mastcam-Z images will provide context and key engineering information in support of the sample location, extraction, and caching, as well as documentation for driving, arm placement, and other rover operations decisions [Bell2020].

Mastcam-Z filter response profiles, including those of the RGB Bayer filters, are described in [Hayes2020] and are shown in [Bell2020].

Mastcam-Z Zoom Motor counts and corresponding Focal Length values are listed in Table 2-9. The relationship of Zoom vs. Focal Length is approximated as a piecewise linear function. Thus for a given Zoom motor count, the corresponding focus  $F$  is defined by:

$$F = \left( \frac{f_2 - f_1}{z_2 - z_1} \right) (z - z_1) + f_1$$

where  $z_1, z_2$  are the min and max values of the range from the table containing  $z$ , and  $f_1, f_2$  are the corresponding min and max focal lengths for the range.

**Table 2-8 – Mastcam-Z Spectral Filters and Bandpass**

Filter Position	Left Eye Wavelength ( $\pm$ Bandwidth), $\lambda_{\text{eff}} \pm \text{HWHM (nm)}$	Right Eye Wavelength ( $\pm$ Bandwidth), $\lambda_{\text{eff}} \pm \text{HWHM (nm)}$
0	630 $\pm$ 43 (Bayer filter Red)	631 $\pm$ 43 (Bayer filter Red)
	544 $\pm$ 41 (Bayer filter Green)	544 $\pm$ 42 (Bayer filter Green)
	480 $\pm$ 46 (Bayer filter Blue)	480 $\pm$ 46 (Bayer filter Blue)
1	800 $\pm$ 9	800 $\pm$ 9
2	754 $\pm$ 10	866 $\pm$ 10
3	677 $\pm$ 11	910 $\pm$ 12
4	605 $\pm$ 9	939 $\pm$ 12
5	528 $\pm$ 11	978 $\pm$ 10
6	442 $\pm$ 12	1022 $\pm$ 19
7	590 $\pm$ 88, ND6	880 $\pm$ 10, ND5



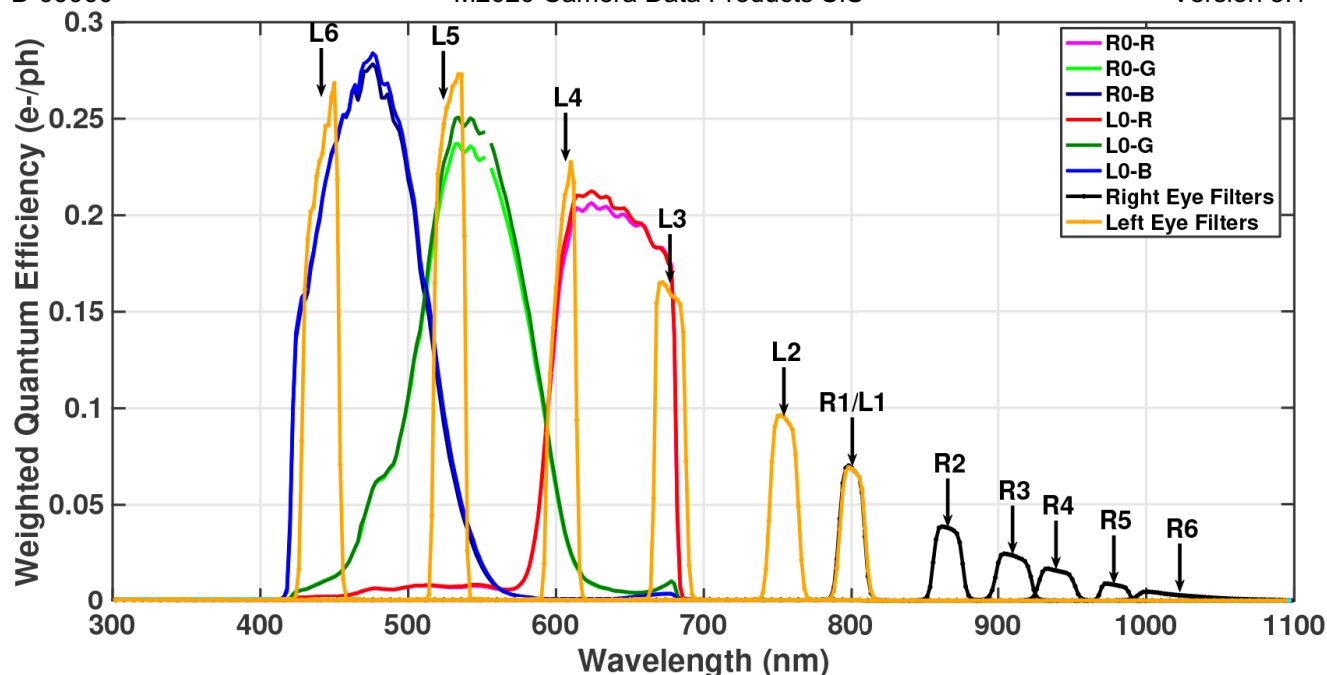


Figure 2-11: Mastcam-Z Spectral Response

Table 2-9 – ZCAM Focal Length to Zoom Motor Counts

Zoom Motor Count (z)	Focal Length (f) (mm)
0	26
2448	34
3834	48
5196	63
6720	79
8652	100
9600	110

## 2.4 SHERLOC Camera Suite

"The SHERLOC instrument is composed of a spectrometer (see [SIS-SHERLOC]) and two cameras, WATSON and ACI. WATSON is used for general observations while ACI captures details of the spectrometer observation."

### 2.4.1 SHERLOC-WATSON

The SHERLOC-WATSON camera is a focusable color camera located on the turret at the end of the M2020 robotic arm and is a build-to-print copy of the MSL MAHLI instrument. The instrument

This document has been reviewed and determined not to contain export controlled technical data.

acquires images of up to 1648 by 1200 pixels (generally only 1600 x 1200 is used) with a color quality equivalent to that of consumer digital cameras using a Bayer pattern. It is also capable of video. SHERLOC-WATSON optics characteristics useful in the analysis of EDR and RDR products are described in Table 2-10. Figure 2-12 shows the locations of both SHERLOC cameras.

Note that the spatial resolution in Table 2-10 measures the working distance. Working distance is measured from the front lens element (front surface of sapphire window) to the target (assumes a planar target parallel to the plane defined by the sapphire window). Working distance can be determined from focus motor count per the formulae (for dust cover close and dust cover open cases) in the Edgett et al. (2019) WATSON Cal Rept. [Edgett2019]

Spatial resolution may be calculated by:

$$PS = 6.7593 + [3.658 \times WD]$$

where PS is the Pixel Scale in  $\mu\text{m}/\text{pixel}$  and WD is the working distance is in cm. Note that the constants above are actuals for the unit onboard the M2020 Spacecraft, and these differ from testbed-WATSON, testbed-MAHLI, and MSL-flight-MAHLI).

**Table 2-10 – SHERLOC-WATSON Operational Characteristics**

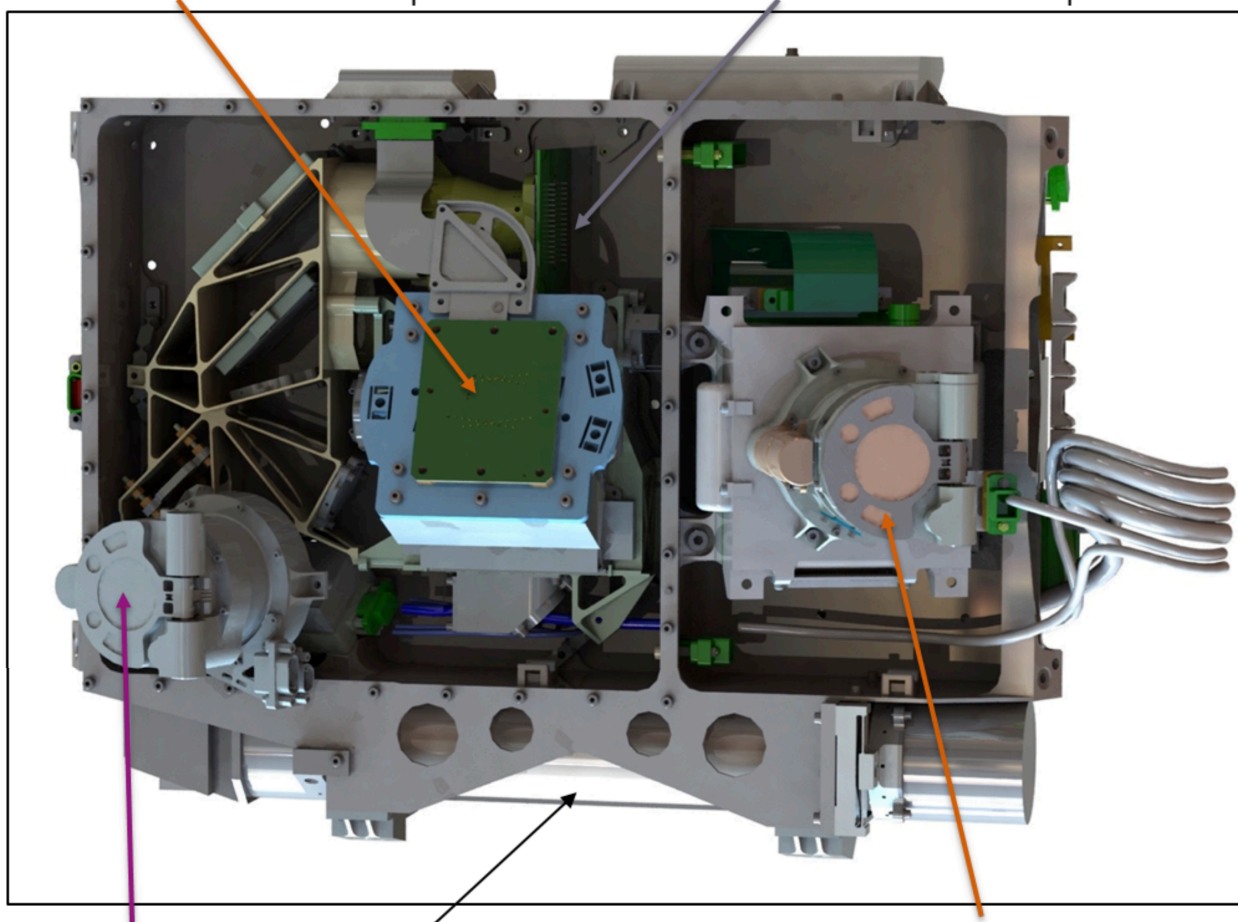
Characteristic	Value
Resolution (S x L)	1648 x 1200 (1600 x 1200 photoactive pixels)
Bit Depth	11
Field of View (FOV)	close range focus = $32.8^\circ \times 24.6^\circ$ infinity focus = $31.2^\circ \times 23.4^\circ$
Spatial Resolution	$\sim 105 \mu\text{m}/\text{pixel}$ at 27 cm distance $15.9 \mu\text{m}/\text{pixel}$ at 25 mm distance
Angular Resolution	0.3 - 0.34 mrad/pixel
Spectral Wavelength $\pm$ Bandwidth ( $\lambda_{\text{eff}} \pm \text{HWHM}$ )	$590 \pm 88 \text{ nm}$ (Broadband)
	$640 \pm 44 \text{ nm}$ (Bayer filter Red)
	$554 \pm 38 \text{ nm}$ (Bayer filter Green)
	$495 \pm 37 \text{ nm}$ (Bayer filter Blue)
Focal Length	18.4 - 21.4 mm
f/number	9.8 - 8.5
Depth of Field	1.6 mm - >4800 mm
Focus Range	17.8 mm - infinity
Number of Spectral Filters	1 plus Bayer pattern on CCD

**SPECTROMETER**

- Asphere-Sphere Spectrometer
- e2v CCD with 512 x 2048 pixels

**CONTEXT IMAGER**

- Monochromatic imager
- Kodak KAI 2020 1200x1600 pixel CCD

**AUTOFOCUS**

- Autofocus mechanism
- Objective element

**LASER**

- 248.6 nm deep UV laser

**WIDE ANGLE IMAGER**

- MAHLI BUILD-TO-PRINT
- LEDs

**Figure 2-12: WATSON(“Wide Angle Imager”) and ACI(“Context Imager”) Cameras of SHERLOC instrument suite.**

## 2.4.2 Autofocus and Context Imager (ACI)

The SHERLOC-ACI is used to capture detailed images of the area observed by the SHERLOC spectrometer. It is co-boresighted with the laser illumination and spectral collection path. Light from the target (indirect sunlight or LED illumination) is collected by the autofocus objective and directed to the context imager by the foreoptics.

Table 2-11 provides the camera characteristics while Figure 2-12 shows the location of the camera.

**Table 2-11 – SHERLOC-ACI Operational Characteristics**

Characteristic	Value
----------------	-------

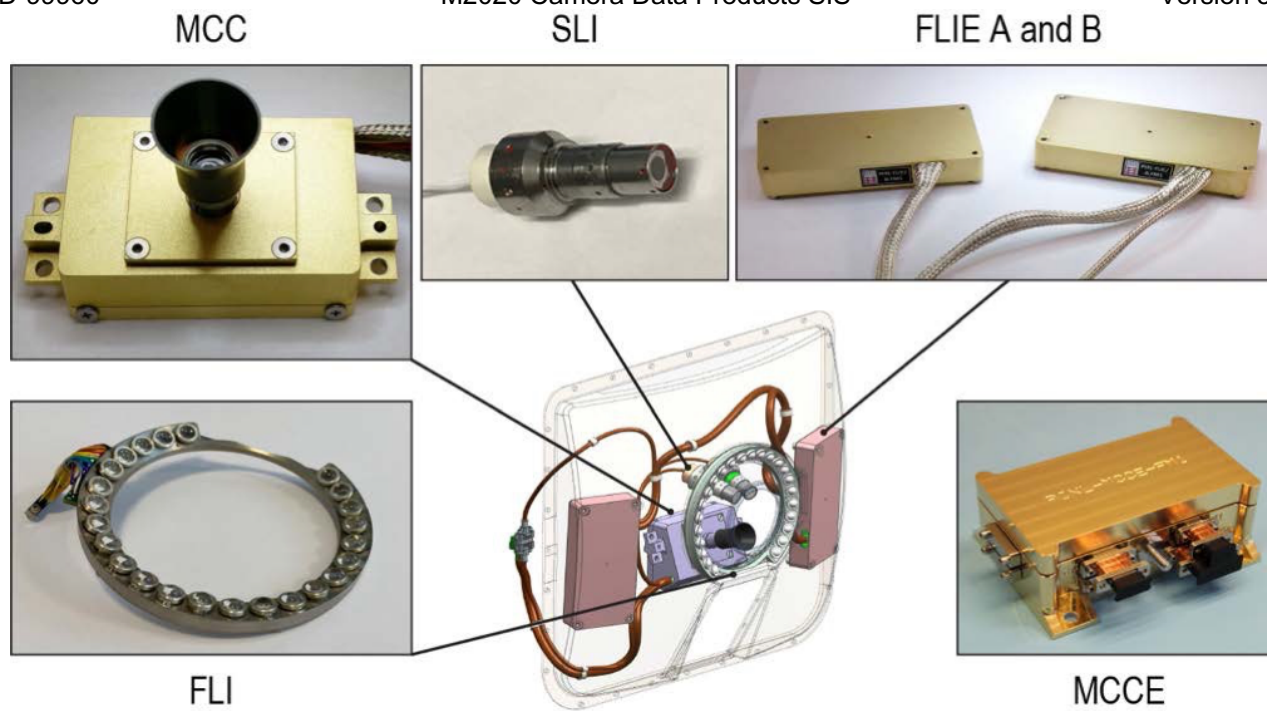
This document has been reviewed and determined not to contain export controlled technical data.

Characteristic	Value
Resolution (S x L)	1648 x 1200 (1600 x 1200 photoactive pixels)
Bit Depth	11
Field of View (FOV)	7.9 x 10.6 degrees 16.16 x 12.12 mm with 11.7mm $\varnothing$ unvignetted
Spatial Resolution	10.1 $\mu\text{m}/\text{pixel}$ at 48 mm distance
Angular Resolution	115 $\mu\text{rad}/\text{pixel}$
Spectral Wavelength	550 $\pm$ 50 nm (grayscale)
Depth of Field	$\pm$ 250 $\mu\text{m}$
Number of Spectral Filters	0 (monochrome detector)
Effective focal length	87.5 mm
Working distance	48 mm

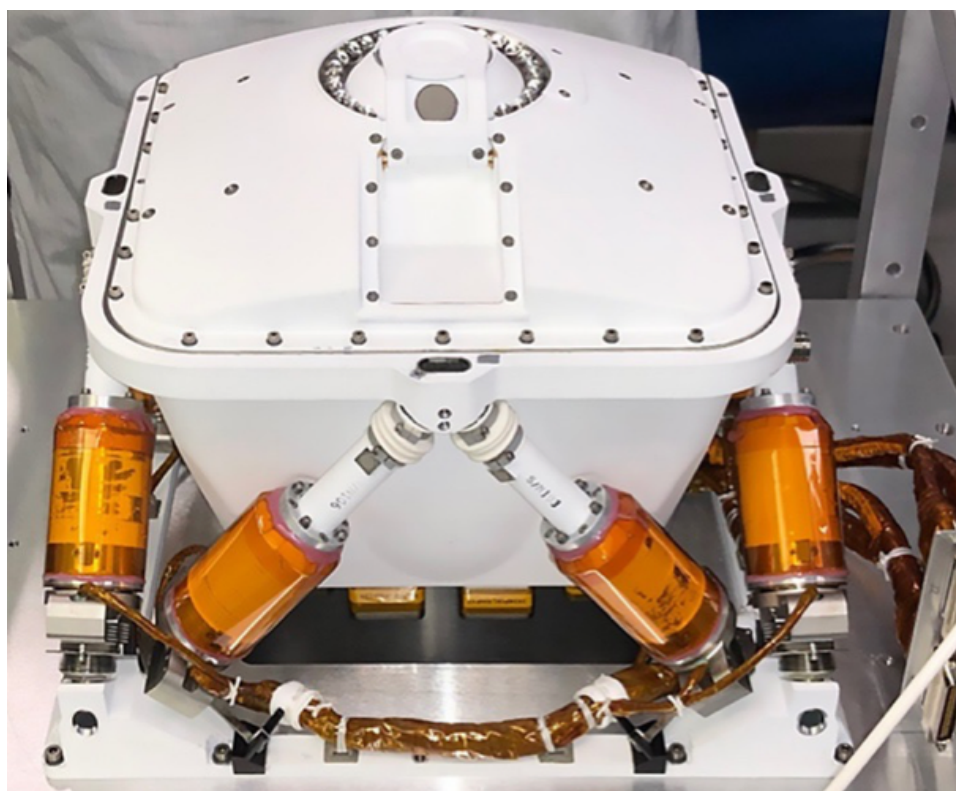
## 2.5 PIXL Micro Context Camera (MCC)

The PIXL Micro Context Camera (MCC) consists of a power and data processing unit, referred to as MCC Electronics (MCCE), the camera unit (MCC) together with the two active light sources: Two Structured Light Illuminators (SLI) and a Flood Light Illuminator (FLI). The SLI is directly driven by the MCC, while the FLI has two external electronics (FLIE) that serve as energy storage and control for the FLI. The MCCE is piggyback mounted to the PIXL electronics in the rover's body, whereas all the optical components are embedded into the PIXL's Sensor Assembly mounted on the turret.

The light sensitive device of the MCC is a CCD with the physical dimensions of 7.95mm x 6.45mm, with 752x580 pixels, each pixel being 8.6 x 8.3  $\mu\text{m}$ . The camera optical lens has an effective focal length of  $\sim$ 9.5 mm. The CCD is tilted 5.2 degrees relative to the boresight in order to maintain an angled depth of focus (Scheimplug Principle), complying with the angled perspective when the PIXL Sensor Assembly is oriented perpendicular to the targeted surface. The FLI has LEDs in 4 color channels Red (735nm), Green (530 nm), Blue (450 nm) and UV (385 nm). Each channel has 6 LEDs and each color can be controlled individually. Furthermore, all the Green LEDs can be individually controlled.

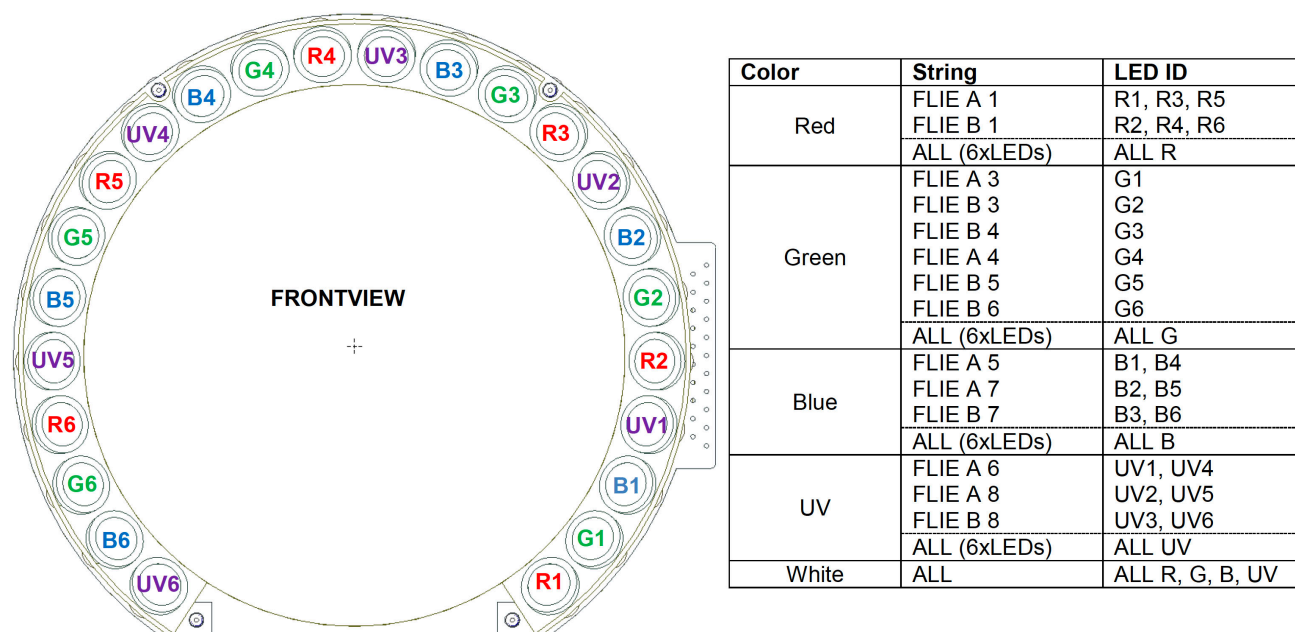


**Figure 2-13: Images of the MCC components and location of sensor head components within the PIXL Sensor Assembly**



**Figure 2-14: The PIXL Sensor Assembly**

Control of whether to use specific color channel, is achieved by setting the current level for each LED string. To disable/enable a specific string, the current is set to one of the four available levels: 0mA, 140mA, 280mA and 500mA. Figure 2-11 provides the mapping of LED IDs on the FLI and an overview of the arrangement of the LED relative to the output strings from FLIE A and FLIE B.



**Figure 2-15: LED distribution on the FLI and corresponding LED ID**

**Table 2-12 – PIXL-MCC Operational Characteristics**

Characteristic	Value
Resolution (S x L)	752 x 580
Bit Depth	8
Field of View (FOV)	37.6 deg x 28.4 deg ~39 mm x 31 mm
CCD Pixel Spatial Resolution	8.6 $\mu\text{m}/\text{pixel}$ x 8.3 $\mu\text{m}/\text{pixel}$
Focal Length	9.54 mm
Angular Resolution	0.91 x 0.87 mrad/pixel
Spectral Wavelength	Continuous in the visible spectrum, reaching from NUV to NIR
Number of Spectral Filters	0 (monochrome detector)

## 2.6 MEDA Sky Camera

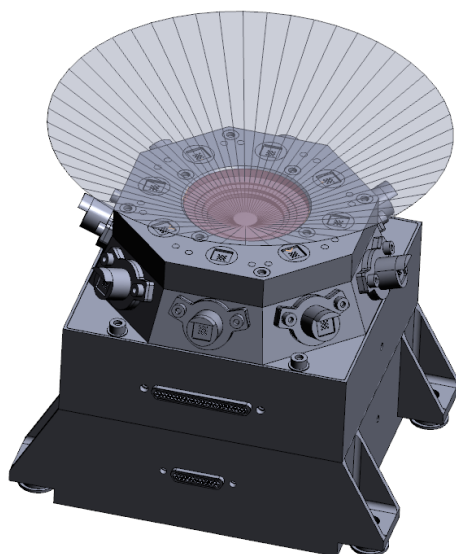
The MEDA Skycam is a fixed camera looking up, which is intended to take images of the sky for weather monitoring purposes. It is a build-to-print copy of the MER and MSL Engineering Cameras (but with a different lens). See Table 2-13 for camera characteristics. The camera is mounted in the



center of the Radiation and Dust Sensor (RDS) and also includes a 120° FOV Baffle (ring on lens) that will be used to block sunlight at specific times during the Martian Sol (Figure 2-16).

**Table 2-13 – SkyCam Operational Characteristics**

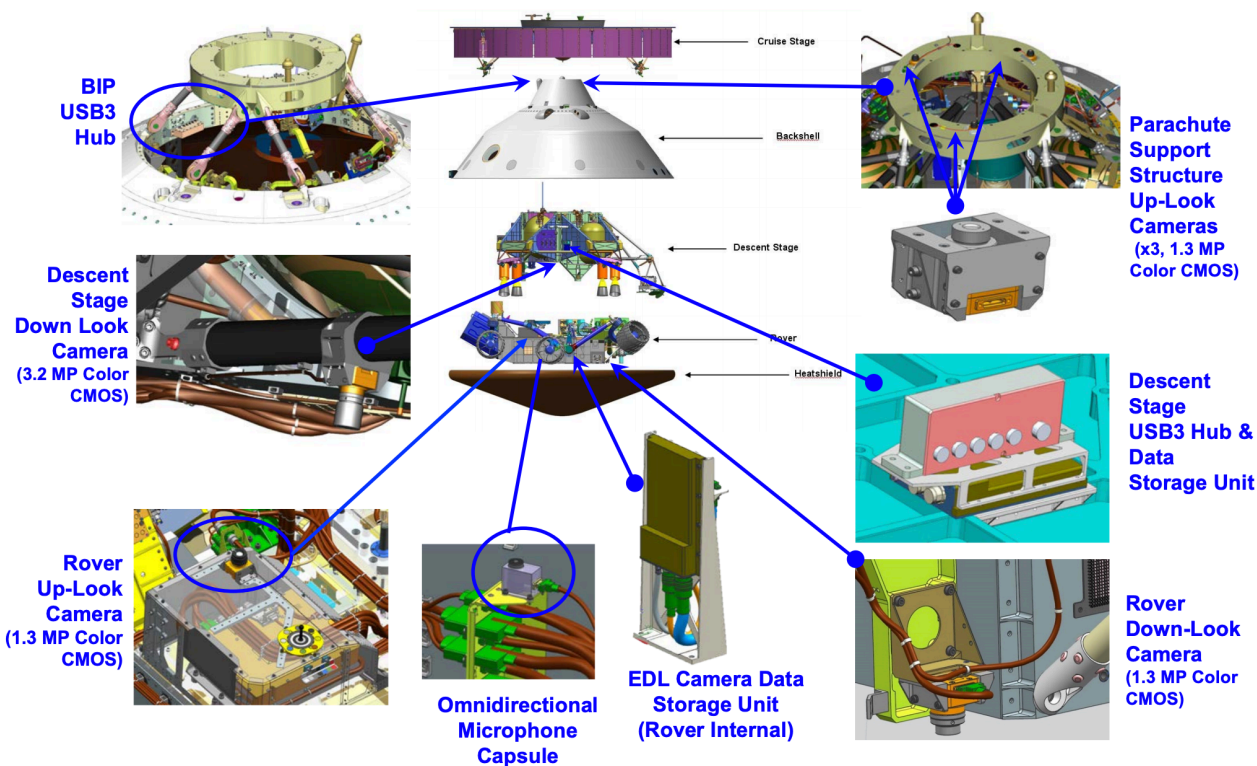
Characteristic	Value
Resolution (S x L)	1024 x 1024
Bit Depth	12
Field of View (FOV)	124 x 124 deg
Angular Resolution	2.1 mrad/pixel at center
Spectral Bandpass	600 - 800 nm
Focal Length	5.58 mm
f/number	15
Depth of Field	0.1 m - infinity
Best Focus	0.5 m



**Figure 2-16: RDS-Skycam with FOV**

## 2.7 EDL Camera Suite

The EDL camera suite consists of 7 COTS FLIR (formerly Point Grey) cameras mounted to the backshell, descent stage structure, and the rover body. They are generally used to capture events during EDL, although the RDC and RUC can be used to observe the surface once on the ground. See Figure 2-17 and Table 2-14.



**Figure 2-17: EDL Cameras overview**

### 2.7.1 Parachute Uplook Cameras (PUC)

The 3 Parachute Uplook Cameras (PUC) are FLIR Chamleon 3 (model CM3-U3-13Y3C-CS). The cameras are mounted on the top of the Backshell. Each camera has identical capabilities and can acquire color (Bayer pattern CCD) images of up to 1280 x 1024 pixels and are capable of video at 149 fps with global shutter readout. The PUCs start to acquire image data immediately before parachute deployment at a frame rate of 75 fps (frames per second) and continue acquiring data at that rate for 30 seconds. After 30 seconds the frame rate drops to 30 fps until backshell separation, expected to occur approximately 98 seconds later. The total number of expected PUC images is ~ 5,190 images per PUC, or 15,570 total images.

### 2.7.2 Descent Stage Downlook Camera (DDC)

This camera is located at the bottom of the Descent Stage structure and will look downward at the Rover to capture imagery of the following events: rover dynamics during bridle descent and mobility deployment and Descent Stage main engine plume interactions with the Martian surface through touch down. The DDC will start acquiring image data just before heatshield separation. The DDC acquires data at 12 fps for ~ 75 seconds and is expected to acquire approximately 900 images.

### 2.7.3 Rover Uplook Camera (RUC)

This camera is located on the top of the Rover and will look upward at the Descent Stage Record descent stage dynamics from sky crane through touchdown and capture flyaway event. The RUC will start acquiring data just before heatshield separation. The RUC acquires data at 30 fps for approximately 130 seconds and is expected to generate approximately 4,000 images.

Note that the RUC is functional after landing and is occasionally used to acquire non-EDL images.



## 2.7.4 Rover Downlook Camera (RDC)

This camera will be located on the side of the Rover and will look downward at the Martian surface during EDL starting at heatshield release through touchdown on the surface. The RDC will start acquiring data just before heatshield separation. The RDC acquires data at 30 fps for approximately 260 seconds and is expected to acquire a total of approximately 7,500 images.

Note that the RUC is functional after landing and is occasionally used to acquire non-EDL images.

## 2.7.5 EDL Microphone

The EDL camera package also includes a microphone, that will be used to record audio during EDL (Entry, Descent, and Landing), and possibly other times throughout the mission.

**Table 2-14 – EDL Cameras Operational Characteristics**

Characteristic	PUC/RUC/RDC	DDC
Resolution (S x L)	1280 x 1024	2048 x 1536
Bit Depth	8	8
Field of View (FOV) (horizontal)	35 deg $\pm$ 3	48 deg $\pm$ 3
Field of View (FOV) (vertical)	30 deg $\pm$ 3	37 deg $\pm$ 3
Pixel scale	~0.5 mrad/pixel	~0.4 mrad/pixel
Focal Length	9.5 mm	8 mm
f/number	f/7	f/5.6
Best Focus	PUC=44 m RUC=8m RDC=infinity	8m

## 2.8 Lander Vision System Camera (LCAM)

The LVS Camera (LCAM) is intended to be used only during EDL to acquire images under high attitude rates and velocities, which are then used by the landing system for navigation. It is located on the outside of the rover body. The camera has a global shutter with a low exposure time and has a wide field of view to be able to capture terrain features even when camera is pointed up to 45° off-nadir. See Table 2-15.

**Table 2-15 – LCAM Operational Characteristics**

Characteristic	Value
Detector Type	Global shutter and grayscale
Resolution (2x2 summed)	1024 x 1024
Bit Depth	8
Field of View	90°x90°
Focal Ratio	f/2.7
Focal Length	5.8mm

This document has been reviewed and determined not to contain export controlled technical data.

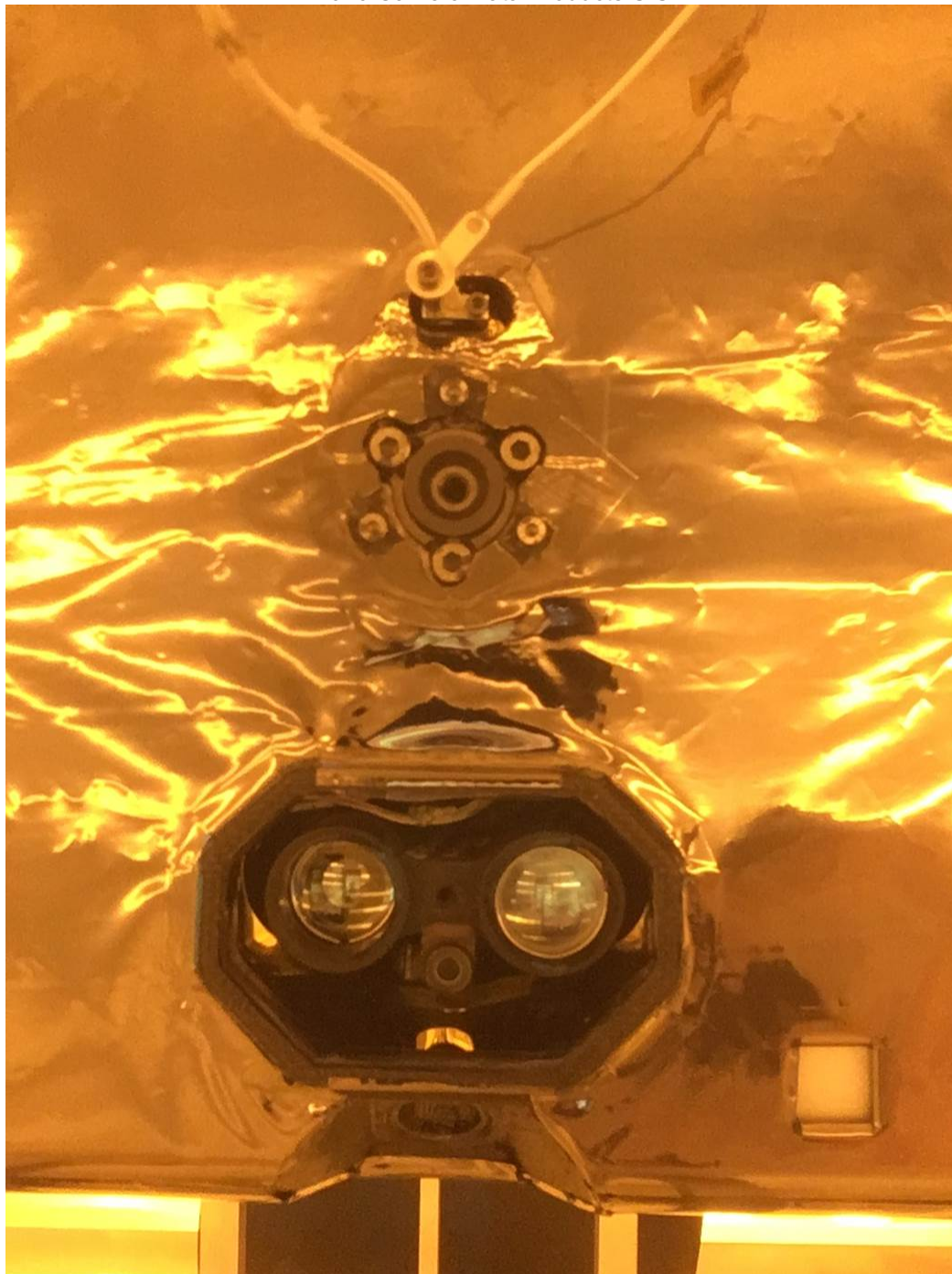
Characteristic	Value
Best Focus	2m to infinity
Signal-to-Noise Ratio	80 at half full well depth
Exposure Time	~1ms
Latency	~100ms

## 2.9 Helicopter imaging suite

The Mars 2020 rover carries a helicopter as a technology demonstration. The helicopter will fly up to 5 times over a ~30 sol period early in the mission. The helicopter carries two cameras, one for navigation and one for capturing aerial views for science evaluation. The images are stored onboard during flight, and then are transmitted to the rover afterwards for relay to Earth.

**Table 2-16 – Helicopter Cameras Operational Characteristics**

Characteristic	Nav	RTE
Resolution (S x L)	640 x 480	4224 x 3120
Bit Depth	8	8
Field of View (FOV)	133 x 100 deg	101 x 82 deg
Angular Resolution	3.6 mrad/pix	0.53 mrad/pix
Number of Spectral Filters	0 (monochrome detector)	0 (color detector)



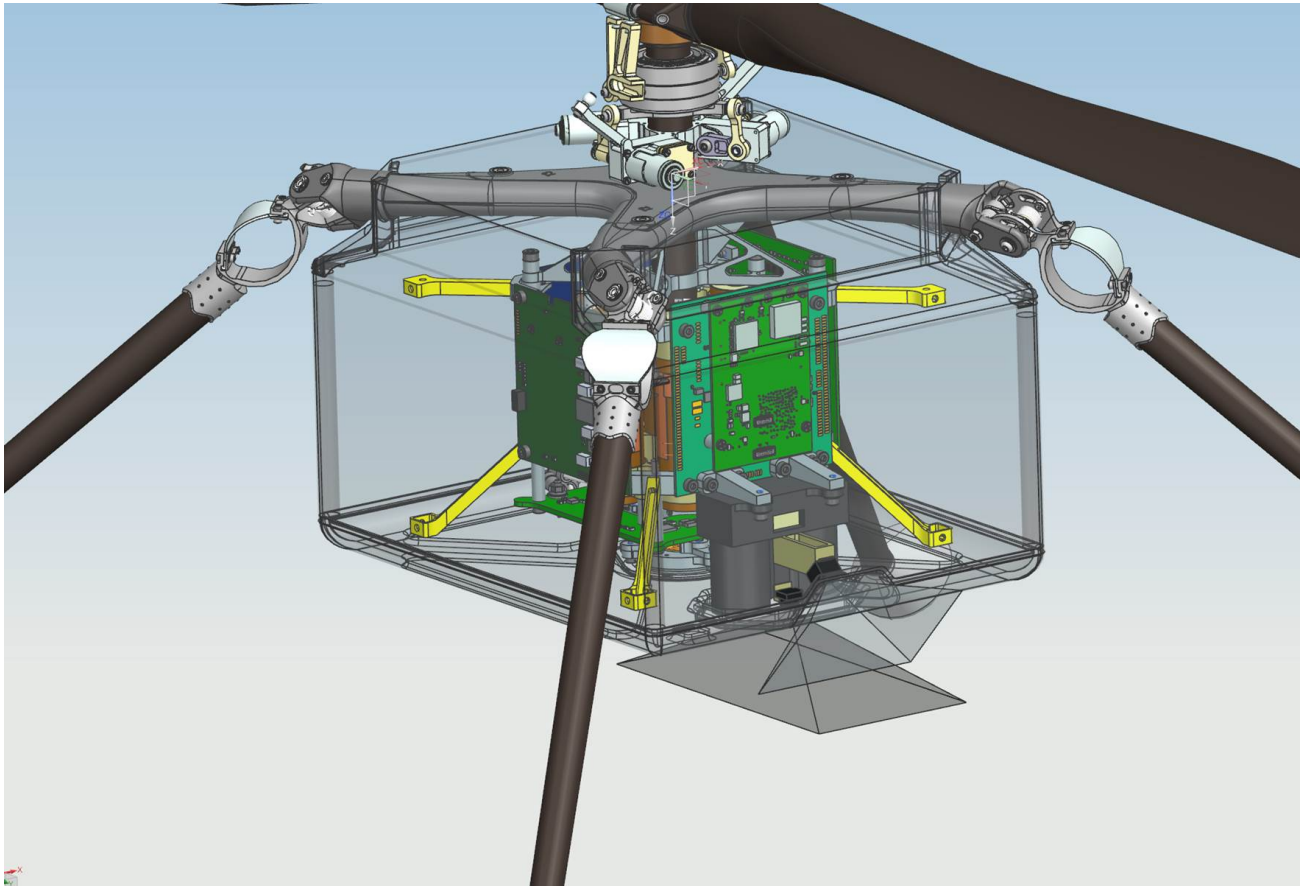
**Figure 2-18: Helicopter Cameras Photo. Bottom View: RTE is at the lower corner in the indentation. Navigation Camera is the "nose" in the oval cutout.**

### **2.9.1 Helicopter Navigation Camera**

The helicopter navigation camera is a 640x480 pixel, 8-bit, grayscale camera. It points straight down from the belly of the helicopter (see Figure 2-19). It is used by the onboard navigation software to determine helicopter position and attitude, and to help navigate to the desired destination. Selected images may be returned to Earth for analysis.

## 2.9.2 Helicopter Return to Earth Camera

The helicopter “Return to Earth” camera is a 4224x3120 pixel, 8-bit, color camera. It is mounted on the side of the helicopter and faces down at a 45 degree angle so it can see both nadir and horizon (see Figure 2-19).



**Figure 2-19: Helicopter Cameras location and viewing frustrums. Navigation Camera looking straight down, RTE angled.**

## 2.10 Camera Flight Software Processing

For certain cameras, onboard processing is used to prepare the data for downlink. There are three types that are worth special mention in this section.

### 2.10.1 JPEG Compression

JPEG data must be processed in two ways before it can be sent. These steps are done in the FSW for the ECAM, RMI, and SkyCam cameras. For ZCAM and both SHERLOC cameras, they are done onboard the camera itself.

First, the data must be converted to 8 bits. This is required because JPEG does not support more than 8 bits. In order to convert from the native 10-13 bit formats, the data is “companded”. This is a form of compression that reduces the dynamic range to 8 bits using either bit shifting or a lookup table (see Section 5.2.2).

Second, the data must be converted to color (see Section 5.2.3). The raw Bayer pattern has a large high-frequency content (since every neighboring pixel is taken with a different filter). This high frequency content would not compress well using JPEG, and would result in serious color artifacts if it were decompressed and then debayered. Therefore, any time JPEG is used for transmission, the Bayer pattern must be removed onboard. This uses the same Malvar algorithm [Malvar2004] as used by the pipeline.

### **2.10.2 Color ICER for SuperCam**

When SuperCam requests sending data in ICER compression mode, the rover FSW extracts the four Red, Green-1, Green-2, and Blue color arrays and sends each array separately to the ICER compressor with the same compression parameters. Rover FSW then packs the 4 separate color channel “images” into a single downlink product. This is re-converted on the ground into the original Bayer-pattern image (which then requires de-Bayering in the ground system). This mechanism limits the damage caused by the potentially lossy ICER compression and enables better compression by removing the very high frequency content imposed by the Bayer pattern itself, while preserving the ability to use other de-Bayering algorithms on the ground.

### **2.10.3 Compression for MEDA**

The MEDA Skycam images stored in instrument flash memory have been compacted into image segments that store 2 pixels of data in 3 bytes of memory. When the instrument transfers the data to the RCE for downlink, the Rover FSW iteratively receives each segment of the image and decompacts the segments to create the full-size image. The full-size image is then compressed per the compression parameters and stored on the RCE for downlink.

### 3. PRODUCT LABEL AND HEADER

Mars 2020 image products consist of at least two files: the image data file in VICAR/ODL format, and a detached PDS4 label file in XML format. Some products also incorporate the original JPEG stream from the spacecraft (EJP type) as a third, supplemental file. The image file is in a dual VICAR/ODL format, the same as was used on MER and MSL. It consists of three major parts: the ODL label, the VICAR label, and the image area. Thus, each image actually has three labels: the attached ODL and VICAR labels, which are embedded within the image data and are used during operations, and the detached PDS4 label, which is used for archiving the data at PDS. The VICAR and ODL labels are very similar (see below). All three labels contain the same semantic content and can be used interchangeably. For products produced by IDS, the VICAR label is produced natively from the tools; the ODL and PDS4 labels are then created from that label, which ensures they have the same semantic content. The VICAR label is used internally by the Mars 2020 project's GDS and also the MIPL software. The ODL label is used by many science teams. Both are included for maximum compatibility, so that operations tools can still be used on archive data.

There are some data products produced by non-IDS teams that do not have the VICAR label, but rather have just the ODL and PDS4 labels. The format is the same, except for the lack of VICAR label.

Figure 3-1 shows the relationships between these labels and the data files for the most common product types.

#### 3.1 ODL Label

As implied in the previous section, the image EDRs and RDRs described in this document have an attached ODL label and a detached PDS4 label. Each institution is responsible for converting PDS-formatted products to be compatible with their own software systems (such as VICAR, IDL, ISIS, etc.).

The attached ODL label starts with the keyword assignment:

ODL\_VERSION\_ID = ODL3

An ODL label is object-oriented and describes the objects in the data file. The ODL label contains keywords for product identification, along with the data object definition containing descriptive information needed to interpret or process the data in the file.

ODL label statements have the form of "keyword = value". Each label statement is terminated with a carriage return character (ASCII 13) and a line feed character (ASCII 10) sequence to allow the label to be read by many operating systems. Pointer statements with the following format are used to indicate the location of data objects in the file:

^object = location

where the caret character (^, also called a pointer) is followed by the name of the specific data object. The location is the 1-based starting record number for the data object within the file. Alternatively, it could be the 1-based byte location within the file if it includes a <bytes> unit tag. Pointers are used to define the locations of the binary instrument data itself (^IMAGE for image data), and the VICAR header in the case of images (^IMAGE\_HEADER).

### 3.1.1 Keyword Length Limits

All ODL keywords are limited to 31 characters in length. Therefore, software that reads M2020 ODL labels must be able to ingest keywords up to 31 characters in length.

For image RDR-producing institutions wishing to accommodate the VICAR mapping of ODL keywords that use a <unit> tag after the value, such keywords must be limited to 25 characters in length to accommodate the “\_\_UNIT” suffix. Otherwise, those keywords will not be transcoded from the ODL label into a VICAR label.

### 3.1.2 Data Type Restrictions

In order to accommodate VICAR dual-labeled files, 16-bit data must be stored as signed data. Unsigned 16-bit data are not supported. 12-bit unsigned data from the cameras are stored in a 16-bit signed value. 8-bit data are unsigned.

### 3.1.3 Interpretation of N/A, UNK, and NULL

During the completion of data product labels or catalog files, one or more values may not be available for some set of required data elements. In this case the literals “N/A”, “UNK”, and “NULL” are used, each of which is appropriate under different circumstances. As a note, if any one of these three symbolic literals are used in place of a keyword value that is normally followed by a Unit Tag(s) (e.g., “<value>”), the Unit Tag(s) is removed from the label.

- “N/A” (“Not Applicable”) indicates that the values within the domain of this data element are not applicable in this instance. For example, a data set catalog file describing NAIF SPK kernels would contain the statement:

```
INSTRUMENT_ID = "N/A"
```

because this data set is not associated with a particular instrument.

“N/A” may be used as needed for data elements of any type (e.g., text, date, numeric, etc.).

- “UNK” (“Unknown”) indicates that the value for the data element is not known and never will be. For example, in a data set comprising a series of images, each taken with a different filter, one of the labels might contain the statement:

```
FILTER_NAME = "UNK"
```

if the observing log recording the filter name was lost or destroyed and the name of the filter is not otherwise recoverable.

“UNK” may be used as needed for data elements of any type.

This document has been reviewed and determined not to contain export controlled technical data.

- “NULL” is used to flag values that are *temporarily* unknown. It indicates that the data preparer recognizes that a specific value should be applied, but that the true value was not readily available. “NULL” is a placeholder. For example, the statement:

DATA\_SET\_RELEASE\_DATE = "NULL"

might be used in a data set catalog file during the development and review process to indicate that the release date has not yet been determined.

“NULL” may be used as needed for data elements of any type.

Note that all “NULL” indicators should be replaced by their actual values prior to final archiving of the associated data.

Unlike earlier missions, some effort has been expended to reduce the number of UNK, N/A, and NULL values appearing in the label, since they can cause difficulties with the Velocity technology used to create PDS4 labels. Therefore, while these values are possible, they should be rare.

### 3.1.4 ODL Label Constructs “Class”, “Object” and “Group”

For the products described in this document, the ODL label includes the following constructs:

- *Class* - The Class construct resides in a ODL label as a grouping of keywords that are thematically tied together. Classes are usually preceded by a label comment, although it is not required. ODL label comments are character strings bounded by “/\* \*/” characters.

In the M2020 Camera ODL label a Class of keywords will be preceded by a comment string as follows:

```
/* comment string */
keyword      = keyword value
keyword      = keyword value
```

- *Object* - The Object construct is a set of standard keywords used for a particular data product. In the M2020 Camera ODL label an Object’s set of keywords is specified as follows:

```
OBJECT          = Object identifier
keyword         = keyword value
keyword         = keyword value
END OBJECT      = Object identifier
```

- *Group* - The Group construct is a grouping of keywords that are not components of a larger Object. Group keywords may reside in more than one Group within the label. In the M2020 Camera ODL label, a Group’s set of keywords is specified as follows:

```
GROUP          = Group identifier
keyword        = keyword value
keyword        = keyword value
END GROUP      = Group identifier
```



### 3.1.5 ODL Image Object

An IMAGE object is a two-dimensional array of values, all of the same type, each of which is referred to as a *sample*. IMAGE objects are normally processed with special display tools to produce a visual representation of the samples by assigning brightness levels or display colors to the values. An IMAGE consists of a series of lines, each containing the same number of samples.

The required IMAGE keywords define the parameters for simple IMAGE objects:

- LINES is the number of lines in the image.
- LINE\_SAMPLES is the number of samples in each line.
- SAMPLE\_BITS is the number of bits in each individual sample.
- SAMPLE\_TYPE defines the sample data type.

The IMAGE object can have a number of keywords relating to image statistics. These keywords are generally not used on Mars 2020. If they are present, they must be updated to reflect the current statistics of the image. Note that the VICAR label never contains these keywords. The statistics keywords are:

- MEAN
- MEDIAN
- MAXIMUM
- MINIMUM
- STANDARD\_DEVIATION
- CHECKSUM

Many variations on the basic IMAGE object are possible with the addition of optional keywords and/or objects. The “^IMAGE” keyword identifies the start of the image.

## 3.2 PDS4 Label

The PDS4 label is a separate file with the same base name and an extension of “.xml”. It is in XML format whose content is controlled by the PDS Information Model and PDS core, discipline, and mission data dictionaries. The PDS4 label contains the same semantic information as the VICAR label, although the format is quite different. For image files (ending in “.IMG”), the label removes the IMG extension and replaces it with “.xml”, in keeping with PDS4 practices. For other files, the “.xml” extension is added to the end of the complete filename, for example “.TXT.xml” or “.obj.xml”. This prevents name collisions, as there are many non-image files that share the same name except for extension. Appendix B contains tables mapping between VICAR/ODL keywords and their corresponding PDS4 construct.

PDS4 class and attribute names make use of a namespace system, of the form “namespace:keyword” where “namespace” is the data dictionary name (such as “img”, “msn”, “geom”, etc) and “keyword” is the class or attribute name. If the namespace is missing, the PDS core data dictionary should be assumed.

## 3.3 VICAR Label

For all image EDR data products and MIPL produced image RDR data products, an embedded VICAR label follows the ODL label and is pointed to by the ODL pointer “^IMAGE\_HEADER”.

This document has been reviewed and determined not to contain export controlled technical data.

The VICAR label is also organized in an ASCII, “keyword = value” format, although there are only spaces between keywords (no carriage return/line feeds as in ODL). The information in the VICAR label is an exact copy of the information in the PDS label as defined in the next section. The reader is referred to the VICAR File Format document for details of the format, which is available at the URL [http://www-mipl.jpl.nasa.gov/external/VICAR\\_file\\_fmt.pdf](http://www-mipl.jpl.nasa.gov/external/VICAR_file_fmt.pdf). The following text is an excerpt which describes the basic structure:

A VICAR file consists of two major parts: the labels, which describe what the file is, and the image area, which contains the actual image. The labels are potentially split into two parts, one at the beginning of the file, and one at the end. Normally, only the labels at the front of the file will be present. However, if the EOL keyword in the system label (described below) is equal to 1, then the EOL labels (End Of file Labels) are present. This happens if the labels expand beyond the space allocated for them. The VICAR file is treated as a series of fixed-length records, of size RECSIZE (see below). The image area always starts at a record boundary, so there may be unused space at the end of the label, before the actual image data starts.

The label consists of a sequence of "keyword=value" pairs that describe the image, and is made up entirely of ASCII characters. Each keyword-value pair is separated by spaces. Keywords are strings, up to 32 characters in length, and consist of uppercase characters, underscores (“\_”), and numbers (but should start with a letter). Values may be integer, real, or strings, and may be multiple enclosed in parentheses (e.g. an array of 5 integers, but types cannot be mixed in a single value). Spaces may appear on either side of the equals character (=), but are not normally present. The first keyword is always LBLSIZE, which specifies the size of the label area in bytes. LBLSIZE is always a multiple of RECSIZE, even if the labels don't fill up the record. If the labels end before LBLSIZE is reached (the normal case), then a 0 byte terminates the label string. If the labels are exactly LBLSIZE bytes long, a null terminator is not necessarily present. The size of the label string is determined by the occurrence of the first 0 byte, or LBLSIZE bytes, whichever is smaller. If the system keyword EOL has the value 1, then End-Of-file Labels exist at the end of the image area (see above). The EOL labels, if present, start with another LBLSIZE keyword, which is treated exactly the same as the main LBLSIZE keyword. The length of the EOL labels is the smaller of the length to the first 0 byte or the EOL's LBLSIZE. Note that the main LBLSIZE does not include the size of the EOL labels. In order to read in the full label string, simply read in the EOL labels, strip off the LBLSIZE keyword, and append the rest to the end of the main label string.

Note that the EOL labels should not appear in archive products.

A binary header may appear in between the VICAR label and the image, containing arbitrary binary data that are not interpreted by VICAR. The number of records in this header is defined by the VICAR system keyword NLB. Binary headers are not used on Mars 2020.

### 3.4 Mapping of ODL and VICAR Labels

In the cases of the attached ODL and VICAR labels, information content is identical, by definition. ODL and VICAR labels may be used interchangeably, for any purpose in the mission. Any software that modifies one label must also modify the other, or remove them entirely. This is often most easily accomplished by stripping off one of the headers, processing the remaining label as desired locally, and then running a conversion tool to re-create the missing header. Such tools will be provided by MIPL.

This document has been reviewed and determined not to contain export controlled technical data.

The mapping between ODL and VICAR keywords is straightforward, and keyword names are usually the same. However, there are some keyword name differences. The mapping rules are as follows:

- For mapped ODL and VICAR (if applicable) keywords, values are identical in all cases with the exception of differences mandated by the file format itself, such as quoting rules. See the respective ODL and VICAR documents for details, but in general, ODL uses double quotes (") while VICAR uses single quotes (').
- For ODL and VICAR label keywords, with the exception of those defining the file format itself (described below), names are identical in both cases.
- Any ODL label group maps 1-to-1 to a VICAR property set with the same name (Group name == property set name). All contained keywords are identical in both cases. The GROUP and END-GROUP keywords are omitted from the VICAR label; PROPERTY keywords are used instead (as per the VICAR file format definition).
- ODL keywords not in a group is identified by an introductory comment (e.g. /\* IDENTIFICATION DATA ELEMENTS \*/). Such classes map 1-to-1 to a VICAR property set. The name of the VICAR property set and the name of the ODL introductory comment map as follows:

**Table 3-1 – ODL Class to VICAR Property Set Mappings**

ODL Class Comment	VICAR Property Set Name
/* FILE DATA ELEMENTS */	special case, see below
/* POINTERS TO DATA OBJECTS */	special case, see below
/* IDENTIFICATION DATA ELEMENTS */	IDENTIFICATION
/* TELEMETRY DATA ELEMENTS */	TELEMETRY
/* HISTORY DATA ELEMENTS */	PDS_HISTORY
/* COMPRESSION RESULTS */	COMPRESSION_PARMS

- For VICAR labels, ODL comments (i.e., /\* *string* \*/) are stored in a keyword named "PDS\_COMMENT". This keyword appears in the VICAR property containing the elements immediately following the comment. When converting from VICAR to ODL, the comment is placed immediately before the group or class. Blank lines should surround the comment. Note that with IDS-generated EDR and RDR data products, multiple comment lines in a Group are not supported.
- The ODL objects IMAGE\_HEADER and IMAGE, as well as the keywords in /\* FILE DATA ELEMENTS \*/ and the ^IMAGE and ^IMAGE\_HEADER pointers (in /\* POINTERS TO DATA OBJECTS \*/) in the table above, do not map directly to VICAR. They all describe the layout of the file and the image data. The VICAR equivalent for all of these items is the VICAR System label. Information maps between these in a straightforward way. It should be trivial to construct a VICAR system label and the above-referenced ODL entities after referring to the respective file-format-definition documents.

This document has been reviewed and determined not to contain export controlled technical data.

Note that the `/* FILE DATA ELEMENTS */` and `/* POINTERS TO DATA OBJECTS */` comments are constant and so are not mapped to PDS\_COMMENT keywords in the VICAR label. They are inserted automatically as part of the system label conversion process.

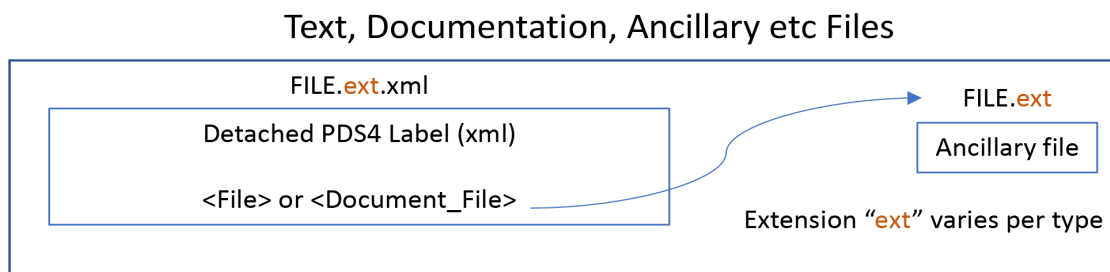
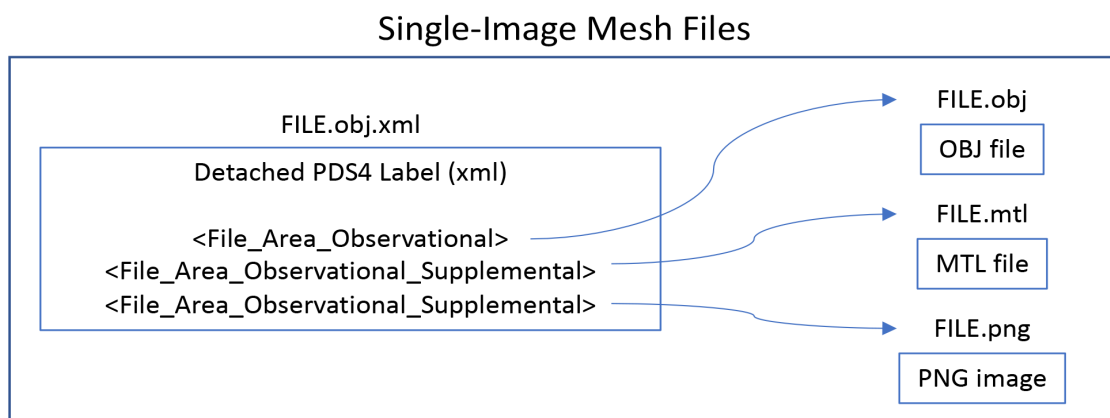
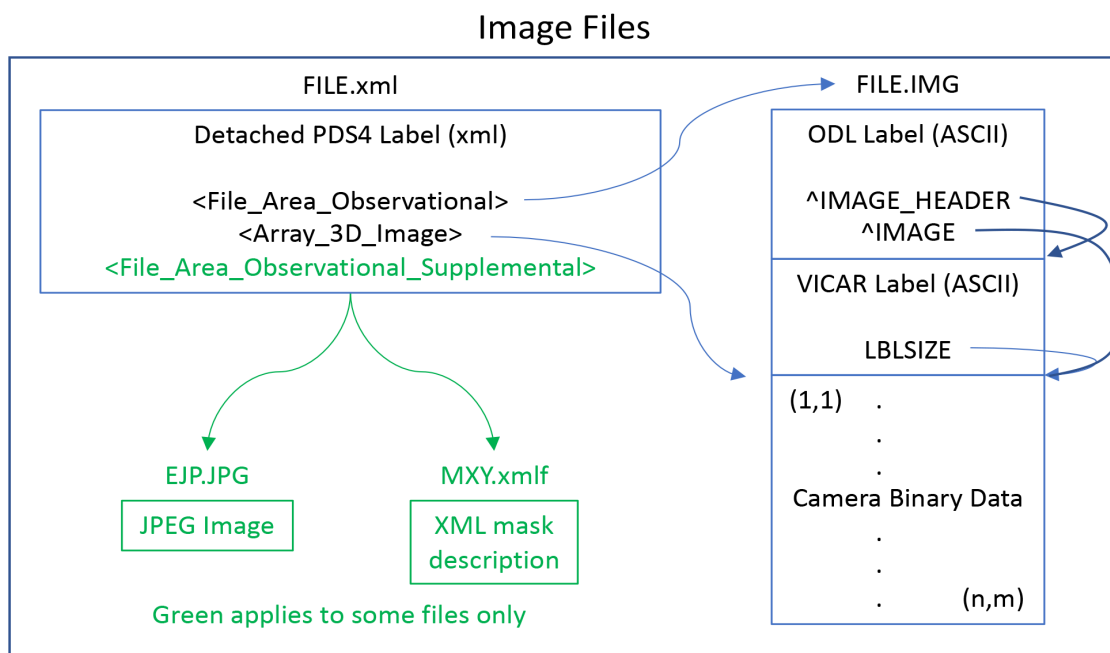
- The statistics-related keywords in the ODL IMAGE object are MEAN, MEDIAN, MAXIMUM, MINIMUM, STANDARD\_DEVIATION, and CHECKSUM. These keywords are never transferred to the VICAR label. For VICAR -> ODL conversion, they can be computed from the image, or simply omitted from the ODL image.
- A few remaining items in the ODL\_IMAGE object are treated specially. The FIRST\_LINE, FIRST\_LINE\_SAMPLE, INVALID\_CONSTANT, and MISSING\_CONSTANT keywords are transferred to the VICAR IMAGE\_DATA property set.
- Any ODL keyword with a `<unit>` tag after the value is transferred to the VICAR label without the unit tag. A VICAR keyword with the same name, but with "`__UNIT`" (two underscores) appended to the end, is added with the value of the unit. So for example, the ODL keyword `"EXPOSURE_TIME = 1.5 <s>"` would translate to two VICAR keywords: `"EXPOSURE_TIME = 1.5"` and `"EXPOSURE_TIME__UNIT = s"`. Note that because of this, any ODL keyword that can support a unit is limited to 25 characters. If there is more than one value (an array), a unit is associated with each. In this case, the "`__UNIT`" VICAR keyword becomes multi-valued also, with each unit copied in sequence. If one of the elements does not have a unit (but others do), the corresponding entry is "N/A" (which is not copied to the ODL label). So for example, ODL `"CONTRIVED_ANGLE = (1.2 <rad>, 22.0, 54.1 <deg>)"` would map to VICAR `"CONTRIVED_ANGLE = (1.2, 22.0, 54.1)"` and `"CONTRIVED_ANGLE__UNIT = (rad, N/A, deg)"`.
- The VICAR history label is omitted from the ODL header.

### 3.5 Mapping of PDS4 and VICAR/ODL Labels

The information contained in the PDS4 and VICAR/ODL labels are semantically equivalent. MIPL uses the VICAR label exclusively for its processing; the PDS4 label is then generated from the VICAR label. This ensures consistency of label contents. In the event that additional VICAR/ODL labels are added during operations but after the PDS peer review, it is possible that those keywords may not appear in the PDS4 label.

It is important to note that these files are simultaneously valid PDS4 images, and valid VICAR/ODL images, and may be processed equally by tools of either system. It is critical for the integrity of the data that both labels be maintained, as described above. Please refer to Appendix B for the mapping between VICAR/ODL keywords and PDS4 attributes.

The conversion from VICAR to PDS4 is accomplished using a Velocity template. Velocity is a transformation engine developed by the Apache Foundation that creates XML documents based on a template. Crucially, it has the ability to embed macros, which are used to extract VICAR label information. The Velocity templates are included with the PDS delivery.



**Figure 3-1: Product Types that have detached PDS4 label**

## 3.6 Label Groups Organization

The Mars 2020 spacecraft, like its predecessors (MSL, MER, et al) uses a command structure where what is requested is not always exactly what is actually acquired. For example, autoexposure or autofocus could be commanded, whereas the returned data has the actual exposure or focus. The command may ask for no motion of the mast, but the returned data contains the actual mast position. Additionally, the camera flight software is somewhat tolerant of command errors, so for example if an illegal subframe is commanded, it will typically take the closest it can get (for example, rounding off to a multiple of 8 pixels, which is required by some cameras) rather than failing the command.

For this reason, the returned metadata often contains an echo of the command parameters, in addition to the result of the command.

In VICAR/ODL, command parameters are in REQUEST groups. These are groups of the form \*\_REQUEST or \*\_REQUEST\_PARMS. Results are the default, they are in groups that do not contain “REQUEST”.

In PDS4, command parameters are contained in a <Commanded\_Parameters> class. This class exists (as of this writing) in the IMG, IMG\_SURFACE, and MSN\_SURFACE dictionaries. The contents of this class consist of attributes and classes that also typically exist outside of a <Commanded\_Parameters> parent. In other words, most of the content of the IMG dictionary is repeated both at the main <Imaging> class, and within <Imaging/Commanded\_Parameters>.

While most commanded items have a corresponding result item, that is not universally the case. In general, look for the non-commanded item to get the actual state of the image, but if it does not exist and the commanded one does, use that.

Another common organizational principle is the “mini-header” group. The mini-header is a small header, typically from the camera itself rather than the rover flight software, that provides a limited set of vital metadata. Mini-headers exist for the following cameras:

- Mastcam-Z: named MINI\_HEADER
- SHERLOC-Watson: named MINI\_HEADER
- SHERLOC-ACI: named MINI\_HEADER
- SuperCam-RMI: named SCAM\_MINI\_HEADER

The first three all have the same layout and keywords, which closely match the MMM cameras in MSL. The SuperCam RMI mini-header is completely different from the others.

Normally, label groups are based on functionality – instrument state, identification, compression, etc. By contrast the mini-header groups are based on where they appear in the telemetry – in the mini-header. This is because these keywords as a set are useful to the respective instrument teams, rather than scattering the items all over the label. However, most of the mini-header keywords are repeated in the functional groups. In general, it is recommended that one look at the functional groups first, resorting to the mini-header group only if the requested information does not exist, or if you are specifically looking for the mini-header.

### 3.7 VICAR/ODL Label Map

This section provides a high-level overview of the VICAR/ODL label, with a basic description of the major groups. This is intended as a guide, not a specification – not all possible groups are shown here, and not all products will contain all groups.

Generally, Mars2020 data products as produced by mission data processing pipelines contain ODL and/or VICAR labels prepended to the primary data object. Both ODL and VICAR group metadata keywords into “paragraphs” or groups of similar keywords. Groups are denoted by “/\*  
GROUP\_NAME \*/” or GROUP=GROUP\_NAME in ODL or by PROPERTY=GROUP\_NAME labels in VICAR. The group structure can be generally mapped to the PDS4 Classes and Attributes (See Appendix B). However, this is far from a 1-to-1 mapping; items from a given group can appear in different PDS4 classes and vice-versa.

The group names vary slightly between ODL and VICAR, see Section 3.4.

**FILE DATA ELEMENTS** - Describes the structure of the data file (similar to PDS4 `File_Area_Observational`). System label for VICAR.

**IDENTIFICATION DATA ELEMENTS** - Contains metadata keywords that identify the mission, target, instrument, timing and specific data collection elements of the data products. These keywords map to many different PDS4 classes throughout the `Product_Observational` PDS4 label, but are generally in `Identification_Area` or the non-discipline parts of `Observation_Area`.

**TELEMETRY DATA ELEMENTS** - Contains metadata keywords that describe how and when the data product was delivered by the spacecraft telemetry stream. These keywords generally map to `msn_surface:Telemetry`.

**HISTORY DATA ELEMENTS** - Contains processing history metadata keywords. These keywords generally map to the `proc:` dictionary. VICAR history is in a different section of the VICAR label but also maps to `proc:`.

**GEOMETRIC\_CAMERA\_MODEL** - Contains keywords defining the camera model for the image. These keywords generally map to `geom:Camera_Model_Parameters`.

**<CS>\_COORDINATE\_SYSTEM** - Contains the definition of the named coordinate system. This group can be repeated for as many coordinate systems as needed. These keywords generally map to `geom:Coordinate_Space_Definition`.

**<OBJECT>\_ARTICULATION\_STATE** - Contains state information for a named articulation device (such as the arm or Remote Sensing Mast). This group can be repeated for as many objects as needed. These keywords generally map to `geom:Articulation_Device_Parameters`.

**OBSERVATION\_REQUEST\_PARMs** – Contains metadata keywords that describe commanded parameters (not results) for a particular observation. These keywords generally map to `Commanded_Parameters` classes in various dictionaries.

**<OBJECT> REQUEST\_PARMS** – Contains metadata keywords that describe commanded parameters (not results) for a particular component of the observation. These keywords generally map to `Commanded_Parameters` classes in various dictionaries.

**INSTRUMENT\_STATE\_PARMS** – Contains metadata keywords that describe the state of the instrument at the time of observation. This is a primary group for result keywords (as opposed to request keywords). These keywords generally map to classes in the `img:` dictionary.

**INITIAL\_STATE\_PARMS** - Contains metadata keywords that describe the initial state of the instrument, specifically, the geometry before Visual Odometry results were applied. These keywords generally map to classes in the `geom:` dictionary.

**COMPRESSION\_PARMS** - Contains metadata keywords that describe the compression performed onboard (for transmission) on the image. Any compression that may occur in the ground processing system is lossless, so is not described in labels. These keywords can generally be mapped to `img:Onboard_Compression`.

**<CS>\_DERIVED\_GEOMETRY\_PARMS** - Contains derived geometry state information for a given coordinate system. This paragraph can be repeated for as many coordinate systems as needed. These keywords can generally be mapped to `geom:Derived_Geometry`.

**DERIVED\_IMAGE\_PARMS** - Contains information about ground processing (image “derived” from the original image). These keywords can be mapped to classes in several dictionaries including `img:`, `img_surface:`, and `geom:`.

**IMAGE\_DATA** - Contains basic information about the image itself. Most of this is in the System label in VICAR, although there is also an `IMAGE_DATA` group in VICAR. These keywords can generally be mapped to the `img:` dictionary and the `File_Area_Observational` Class.

**SURFACE\_PROJECTION\_PARMS** – Describes the map projection used for mosaics. These keywords generally map to classes in the `cart:` dictionary.

**SURFACE\_MODEL\_PARMS** – Describes the surface model used for map projection of mosaics. These keywords generally map to classes in the `cart:` dictionary.

## 3.8 PDS4 Label Map

This section provides a high-level overview of the PDS4 label, with a basic description of the major classes. This is intended as a guide, not a specification – not all possible classes are shown here, and not all product labels will contain all classes.

In general, PDS4 labels are constructed with the `PDS4 Product_Observational` Class, a sub-class of `Product`. The `Product_Observational` sub-class describes a set of information objects produced by the camera system. Other types of PDS4 `Product` Classes follow similar structures but will not map one-to-one with the example presented here.

`Product_Observational`

`Identification_Area` - identify and name an object.

This document has been reviewed and determined not to contain export controlled technical data.



`logical_identifier` - name/location of file

`version_id` - version of product

`title` - Name of file

`information_model_version` - version of PDS4 information model used to create product

`Observation_Area` - information about the circumstances under which the data were collected.

`Time_Coordinates` - time at which data was acquired

`Primary_Results_Summary` - high-level description of the types of products included in the collection or bundle

`Investigation_Area` – describes mission, observing campaign or other coordinated, large-scale data collection

`Observing_System` – describes the observing system (rover and instrument)

`Target_Identification` – identifies the target

`Discipline_Area` – discipline specific information. This is the meat of the label.

`cart:Cartography` – contains mosaic map projection parameters

`disp:Display_Settings` – describes how to view the image

`geom:Geometry` – describes the geometry of the image. Includes the camera model, coordinate system definitions, and states of movable devices.

`img:Imaging` – describes how the image was acquired and processed, and characteristics of the image

`img_surface:Surface_Imaging` – extra imaging parameters relating specifically to surface missions

`msn:Mission_Information` – information regarding the mission generally.

`msn_surface:Surface_Mission_Information` – information specific to surface missions. Includes telemetry and command execution information.

`msss_cam_mh:MSSS_Camera_Mini_Header` – contains the mini-header for Mastcam-Z and both SHERLOC cameras

`proc:Process` – contains processing history

`Mission_Area` - mission specific attributes needed to describe data product

`mars2020:Mars2020_Parameters` – Mars 2020 specific classes and attributes

`mars2020:PIXL_Parameters` – information about the PIXL illuminators

`mars2020:SuperCam_Parameters` – SuperCam RMI mini-header

`Reference_List` – contains pointers to source products and browse images

`File_Area_Observational` - describes the actual data tables

`File` - identifies the file that contains one or more data objects as described below.

`Header` – describes the VICAR/ODL labels.

`Table_Delimited` – Describes CSV tables (not used for images, used for meshes).

`Array_2D_Image` – describes the actual image array (single band)

`Array_3D_Image` – describes the actual image array (multiple bands)

`File_Area_Observational_Supplemental` - describes a supplemental data file, such as an EJP or FITS file, which is related to but is not the primary science data.

## 3.9 SuperCam FITS files

SuperCam additionally stores EDR images in FITS format. These images have the same image content as the VICAR/ODL files. In addition, the metadata is stored in HDU (Header Data Unit) structures within the FITS file. It consists of instrument State of Health (SOH), autofocus, and relevant instrument information, in binary format as received in the telemetry stream. A copy of the VICAR label is also stored in an HDU.

All SuperCAM FITS EDRs follow the same scheme for HDU 0 to 4:

- The primary HDU (HDU#0) contains only a FITS header, which consists of a number of cards in the format “key: value, comment”, which is the standard format for FITS headers. These cards are meant to give rapid access to useful information such as product creation, metadata, or to the data itself. Some cards are common to all EDRs (e.g. creation time, FITS\_GEN version used), and some depend on the EDR type (number of shots for a LIBS product, exposure time for RMI ...)
- HDU 1 contains a copy of the ODL or VICAR label that was injected at the time of the FITS creation, in the form of a binary table.
- HDU 2 contains a table with all the I-cmds (and their arguments) and data replies that were recorded in the generic buffer of the BU. It is meant to be the timeline of cmds and replies that are associated with the commands that produced the file.
- HDU 3 and 4 contains the command reply to MU\_SEND\_HOUSEKEEPINGS and COLLECT\_SOH I-commands, respectively. Each SOH (according to the command dictionary) is reported in one separate column, and each line represent the command reply at a given time. The millisecond-offset to the last TIMESYNC is reported for each line, followed by numeric SOH (engineering units), numeric SOH converted to physical units using the appropriate transfer functions (when appropriate).

It is worth mentioning that a single data product can spawn up multiple EDRs (e.g. a DP containing an RMI and an autofocus (AF) will spawn two FITS EDRs: one for the RMI and one for the AF. EDRs that are spawned from the same DP will have identical HDU 1, 2, 3 and 4. The content of HDU 5 and above depend on the science data that is stored in the file. For RMI EDRs, these HDUs contain:

1. ICER/RAW images: R, G1, G2, B images (2D arrays of 1024x1024 pixels for nominal ROI).
2. JPG images: R, G, B images (2048x2048 pixels for nominal ROI)

More details on the format and data content of the FITS files is given in an appendix of the non-imaging SIS [SIS\_SCAM].

## 4. GENERAL DATA PRODUCT OVERVIEW

### 4.1 Data Processing Levels

This documentation generally avoids the PDS4 processing level terminology, instead using the EDR/FDR/RDR category terminology adopted by the Mars 2020 operations team. Table 4-1 shows the general mapping between the two. EDRs are defined as products that reconstruct as closely as possible the data acquired by the camera, given transmission constraints. So they include the "Raw" products derived straight from telemetry, as well as the "Partially Processed" products relating to decompanding and de-Bayering, which simply reconstruct the data acquired by the sensor without any form of calibration or processing. FDRs are the "Fundamental Data Record"; this is a consistently-formatted product at the end of the EDR chain that is used as the basis for all downstream RDR processing (it also includes tile reassembly for the Engineering Cameras, as well as some label updates on certain instruments). RDR (Reduced Data Records) are then all downstream products, encompassing the PDS levels "Partially Processed", "Calibrated", and "Derived".

Table 4-1 lists the PDS4 processing levels and the general categories they apply to, while Table 17-2 shows the PDS4 processing level for each specific type of image.

**Table 4-1 – PDS4 Processing Levels for Instrument Experiment Data Sets**

Processing Level for PDS4 Archive	Operations Data Product Name	Description
Telemetry	n/a	An encoded byte stream used to transfer data from one or more instruments to temporary storage where the raw instrument data will be extracted. PDS does not archive telemetry data.
Raw	EDR (Experiment Data Record, heritage term based on MSL mission)	Original data from an instrument. If compression, reformatting, packetization, or other translation has been applied to facilitate data transmission or storage, those processes will be reversed so that the archived data are in a PDS approved archive format. For Mars 2020, these are the ECM/ECV/ECZ/EDM/ECR original products.
Partially Processed	EDR FDR (Fundamental Data Record) RDR (Reduced Data Record)	Data that have been processed beyond the raw stage, but which have not yet reached calibrated status. For Mars 2020, these are the decompanded and debayered EDRs, the FDRs, and a few RDRs.
Calibrated	RDR	Data converted to physical units, which makes values independent of the instrument.
Derived	RDR	Results that have been distilled from one or more calibrated data products (for example, maps, gravity or magnetic fields, or ring particle size distributions). Supplementary data, such as calibration tables or tables of viewing geometry, used to interpret observational data should also be classified as "derived" data if not easily matched to one of the other three categories.

## 4.2 Binary Data Storage Conventions

EDR and RDR data products for M2020 camera image are stored as binary data. For the image EDRs, the data formats include rescaled 8-bit integers stored in an unsigned byte, as well as 10-bit to 13-bit integers stored in signed 16-bit integers. The PDS and VICAR labels are stored as ASCII text.

### 4.2.1 Bit and Byte Ordering

The ordering of bits and bytes is only significant for instrument and binary header data; all other labeling information is in ASCII.

For non-byte instrument data, which includes 8-bit unsigned shorts, 16-bit signed shorts, 32-bit signed ints, and 32- and 64-bit IEEE floating-point numbers, the data may be stored in either Most Significant Byte first ("big-endian", as used by e.g. Sun computers and Java), or Least Significant Byte first ("little-endian", as used by e.g. Linux and Windows computers). In a EDR/RDR product, the instrument data can have only one ordering, but it is dependent on the host platform where the data was processed. This follows both the ODL and VICAR file format conventions. See Table 4-2.

For all image data, the ODL label carries keyword `SAMPLE_TYPE` in the `IMAGE` Object to define which ordering is used in the file. The VICAR label carries keywords `INTFMT` and `REALFMT` in the `System` portion of the label to define the ordering. Both of these file formats specify that bit 0 is the least significant bit of a byte.

**Table 4-2 – M2020 Image EDR/RDR Bit Ordering**

Address	MSB-first	LSB-first
n	most significant byte	least significant byte
n+1	next	next
n+2	next	next
n+3	least significant byte	most significant byte

## 4.3 Time Conventions

The spacecraft clock is the onboard time-keeping mechanism that triggers most spacecraft events, such as shuttering of a camera. Since telemetry data are downlinked with this clock's time attached to it, spacecraft clock time (SCLK-pronounced "s-clock") is the fundamental time measurement for referencing many spacecraft activities. On M2020 mission this SCLK consists of two fields in 10+5 format with '.' serving as separator: SSSSSSSSSS.FFFFF

where:

SSSSSSSSSS – count of on-board seconds since the spacecraft clock start

FFFFF – count of msec or count of fractions of a second with one fraction being 1/65536 a second. (see below)

Note that the fractional field is in milliseconds for all reported SCLK values, with the sole exception being `COMMAND_DISPATCH_SCLK` where in the Label, it is a count of fractions of a second with one fraction being 1/65536 of a second. In the PDS label this is indicated by ':' rather than '.',

This document has been reviewed and determined not to contain export controlled technical data.

in VICAR/ODL label, it's still '.' Also note that `COMMAND_DISPATCH_SCLK` value is used in the filename for Mastcam-Z products, where fraction of a second value is converted to milliseconds. Thus in the filename of all data products, it is always milliseconds.

SCLK is measured from an epoch time of January 1, 2000, 12:00:00 Ephemeris Time, commonly called J2000. This is the beginning of Julian Year 2000, and corresponds to a Julian date of 2451545.0. This is equivalent to January 1, 2000, 11:58:55.816 UTC. Ideally, the SCLK should equal the exact number of seconds since the epoch. Deviations from this ideal will occur due to clock drift and errors in setting the SCLK value.

SCLK may be recorded in the filename of data product or written as a value of label keyword (see Appendix F) or it could be converted to other time keeping standards listed below before being written into the label:

Spacecraft Event Time (SCET): commonly used to mark a specific event such as science observation

Local Mean Solar Time (LMST): timekeeping that factors in the mean sun's movement.

Local True Solar Time (LTST): timekeeping that factors in apparent (true) sun's movement

Coordinated Universal Time (UTC): The worldwide standard of timekeeping.

Earth Received Time (ERT): includes one-way light time for radio signal to reach Earth from Mars.

## 4.4 Rover Motion Counter (RMC)

The Rover Motion Counter (RMC) is a set of indices whose values serve to uniquely identify every location the rover or vehicle has visited (at some level of quantization), and when any mechanism has moved.

There are three separate RMC namespaces on the Mars 2020 mission: Rover surface ops, Helicopter and LCAM. All use the name "Rover Motion Counter" as a generic term, although "Motion Counter" might be more accurate in the case of the helicopter. Software heritage drives us to use RMC for all three instances (it was called "Rover Motion Counter" on the Phoenix and InSight missions as well, despite them being stationary landers).

In general, an RMC consists of a set of counters, each of which corresponds to either a vehicle location or a bit of hardware that can move. When the vehicle reaches a new location, or the hardware moves, the corresponding RMC counter increments. The index names for each of the RMC namespaces are shown in Table 4-3.

It is critical to pay attention to the RMC index names in the label, to distinguish between these three namespaces.

Index	Rover Surface	Helicopter	LCAM
1	SITE	FLIGHT	SET
2	DRIVE	POS	INSTANCE
3	POSE		
4	ARM		
5	SHA		
6	DRILL		
7	RSM		
8	HGA		
9	BITCAR		
10	SEAL		
11	RTT (PIXL only)		
12	PMC (PIXL only)		

#### 4.4.1 Rover Surface Ops RMC Namespace

This RMC is entirely analogous to the RMC used on MSL and MER. It consists of 10-12 indices (10 in most cases, 12 for PIXL). The first three represent vehicle locations, while the rest represent mechanism locations.

##### 4.4.1.1 Site Index

The Site index defines which instance of the Site frame is relevant for this RMC (see Section 20.5). Whenever the Site is incremented, all the other RMC values are set to 0. Declaring a new Site thus resets all motions and creates a new local area in which to work.

Unlike the other indices, there is no meaning to odd or even values. Also unlike other indices, Sites start with 1 (the landing site), not 0.

##### 4.4.1.2 Drive Index

The Drive index (sometimes called Position) increments whenever the rover drives or otherwise moves its wheels (e.g. trenching or steering). The value is odd while the wheels are actually moving, and even otherwise. Incrementing Drive sets all the other indices (except Site) to 0. Exception: if a mechanism index is odd, indicating it is moving during the drive, then that index is not reset to 0. This should be an unusual case, however. The Drive index is the one of most interest for most localization activities

##### 4.4.1.3 Pose Index

The Pose index indicates a change in the rover's knowledge of its position or orientation. Unlike the other indices, it does not indicate that anything actually moved; only that the pose *knowledge* has changed. This may be due to running visodom (visual odometry), doing a sun find, reading the IMU's (Inertial Measurement Units), or receiving an explicit ground command to update the pose knowledge. Note that the rover might actually have moved slightly, mostly in orientation, due to being pushed by the arm (e.g. during a drill preload). There is unfortunately no reliable way to

This document has been reviewed and determined not to contain export controlled technical data.

distinguish this from an attitude update, although the arm index will change during arm movements. So a pose change does not indicate arm-induced motion, but lack of an arm update does rule it out.

Pose will be odd while an update is actually being made (including extended IMU updates), even otherwise. Incrementing Pose has no effect on other indices.

#### **4.4.1.4 Mechanism Indices**

The remaining RMC indices (Arm, SHA, Drill, RSM, HGA, BITCAR, Seal) are associated with mechanisms on the rover. They increment to an odd number when the mechanism is moving, and again to an even number once the mechanism stops. More than one mechanism may be moving at the same time.

The Mars 2020 arm has the strength and mass to move the rover (in tilt, primarily) when it is used. Thus the fact that the arm moved, as reflected in the RMC, means that the rover's physical pose may have changed, even if its knowledge has not. This is sometimes relevant and sometimes not, but is the primary reason for maintaining the mechanism RMC indices. The other mechanisms could in theory similarly move the rover, but are unlikely to in practice.

### **4.4.2 Helicopter RMC Namespace**

The helicopter is treated as an independent vehicle from the rover; thus, it has its own separate RMC namespace. The 2 elements of the helicopter namespace are Flight and Pos.

#### **4.4.2.1 Flight Index**

The Flight index for the helicopter is entirely analogous to the Site index for the rover. The first flight was flight 1, but there is some "flight" 0 data for checkout imaging activities before flights started. Images taken between flights (on the ground) are labeled with the prior Flight number (see Section 5.6.9.1). Incrementing the Flight index resets the Pos index.

#### **4.4.2.2 Pos Index**

The Pos index for the helicopter is analogous to the Drive index for the rover. It increments during the flight to represent positions at which images were taken. It increments only when an image was taken.

Unfortunately, due to implementation limitations, the Pos counter does not strictly increment on a time basis. It increments separately for the set of Navcam images and the set of RTE images. It can thus be used to identify individual helicopter locations but not the global ordering. The SCLK can be used for image ordering.

For both Navcam and RTE, the Pos index is the count of that type of image taken this flight (starting at 0). Navcams are taken every 1/30 second for navigation reasons; only a few of these are sent to Earth. The Pos count for Navcam is thus the time since the flight started, in units of 1/30 second. For RTE, the count is again incremented when images are taken, but this is only when commanded and is not regular. Thus the RTE count is much lower, and every RTE is usually downlinked. There is thus very little probability that the two will conflict.

Note that due to a bug, the RMC(2) field in the filename, which should be the Pos index, is hardcoded to 1. The Pos index appears in the sequence ID field of the filename, as well as in the Rover Motion Counter field in the label. The RMC(1) field in the filename is the Flight number.

### 4.4.3 LCAM RMC Namespace

The LCAM images have a separate RMC, because they were taken before landing, thus the traditional site/drive does not apply. The RMC indices are Set and Instance. The Set number is based on the sequence ID but was always 0 as flown. The Instance number is the sequence number of LCAM images, starting at 1. Each LCAM image thus has a unique instance number, which is analogous to the rover Drive counter.

### 4.4.4 RMC and Coordinate Frames

The RMC can be thought of as a clock, albeit one that ticks at irregular intervals. It measures time not as a duration, but in terms of motions.

Given any two valid RMC's, one can determine unambiguously which occurred earlier in time, because the counters only count upwards (except when reset to 0 by incrementing Site/Flight/Set or Drive). If the Sites/Flights are different, the lowest is earlier. If the same, then the lower Drive/Pos is earlier (except as noted above, where the navcam and RTE are numbered separately for Pos). If those are the same, then consider the remaining indices. If any are lower, then the entire RMC is earlier. Note that it is not valid for some of these indices to be lower and others higher - that indicates the RMC value was corrupted, possibly during transmission.

The RMC is often used as the index to identify different instances of a coordinate frame (see Section 20.1). The RMC identifies the moment in time at which the coordinate frame is defined. For example, if the rover is at Site 5 drive 24, the Rover(5,24) frame is defined to be coincident with the rover position when the rover was at RMC (5,24). There is thus a strong relationship between the RMC and coordinate frames, but they are distinct concepts. The RMC marks a moment in time; there are coordinate frames whose definitions are tied to that moment in time. But an RMC is not a coordinate frame.

In the Rover namespace, RMC values of (site=n) are associated with instances of Site frame, and RMC values of (site=n,drive=m) are associated with Rover Nav, Rover Mech, or various other frames. RMC values of (rsm=n) are associated with instances of RSM frame, etc.

In the Heli namespace, RMC values of (flight=n) are associated with instances of Heli G frame, and RMC values of (flight=n,pos=m) are associated with instances of Heli M, S1, and S2 frames.

In the LMC namespace, RMC values of (set=0,instance=n) are associated with instances of Cint frame.



## 5. EDR PRODUCT SPECIFICATION

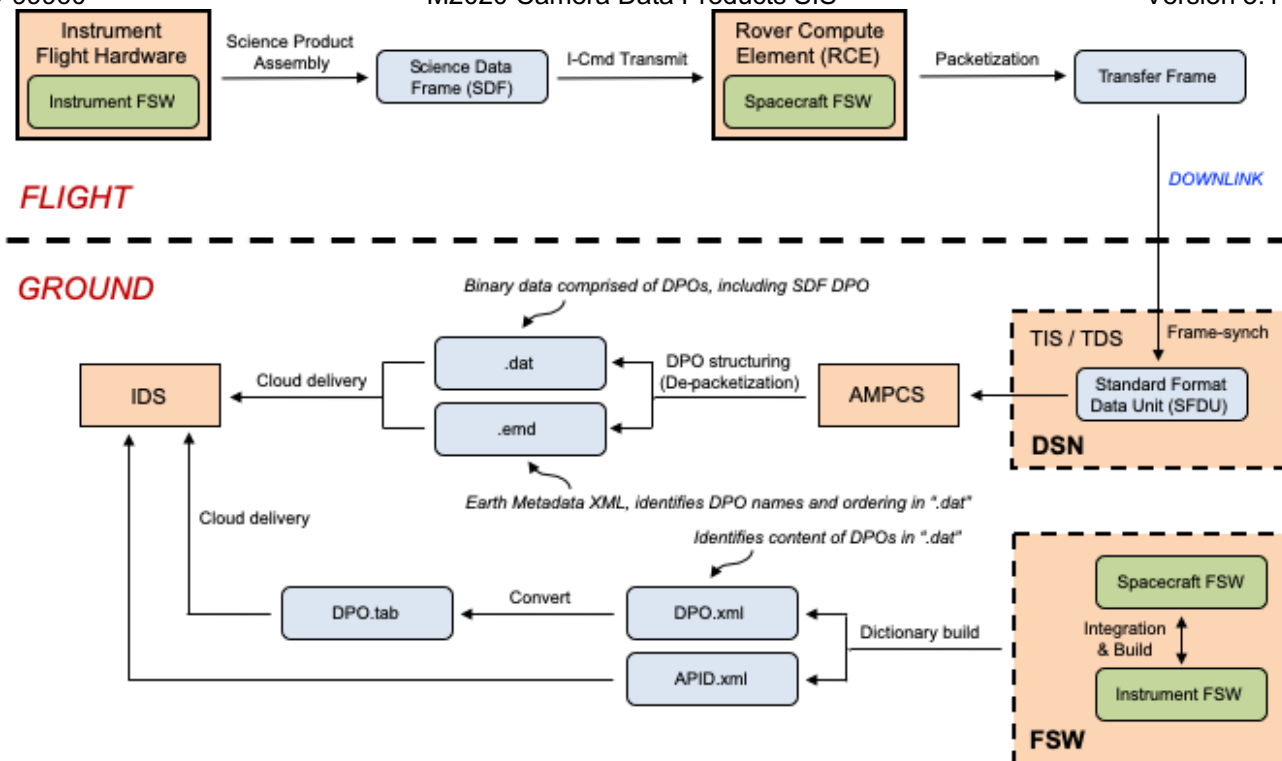
M2020 camera EDRs described in this document will be generated by JPL's Multimission Instrument Processing Laboratory (MIPL) under the IDS subsystem of the M2020 GDS element.

The goal of a camera EDR is to reconstruct, as closely as possible, the data as it appeared on the image sensor. Generally speaking, that means that compression done in order to transmit the data to the ground is reversed in the EDRs. On M2020, EDR is a category, encompassing three sub-categories: Raw, Decompanied, and Debayered. Within each sub-category, there are several image types. See Section 5.2.

EDRs are versioned on M2020. When updated telemetry is received a new version of the EDR will be generated. Updated telemetry is possible when missing packets are retransmitted and/or telemetry is reflowed through the Ground Data System (GDS).

### 5.1 EDR General Processing

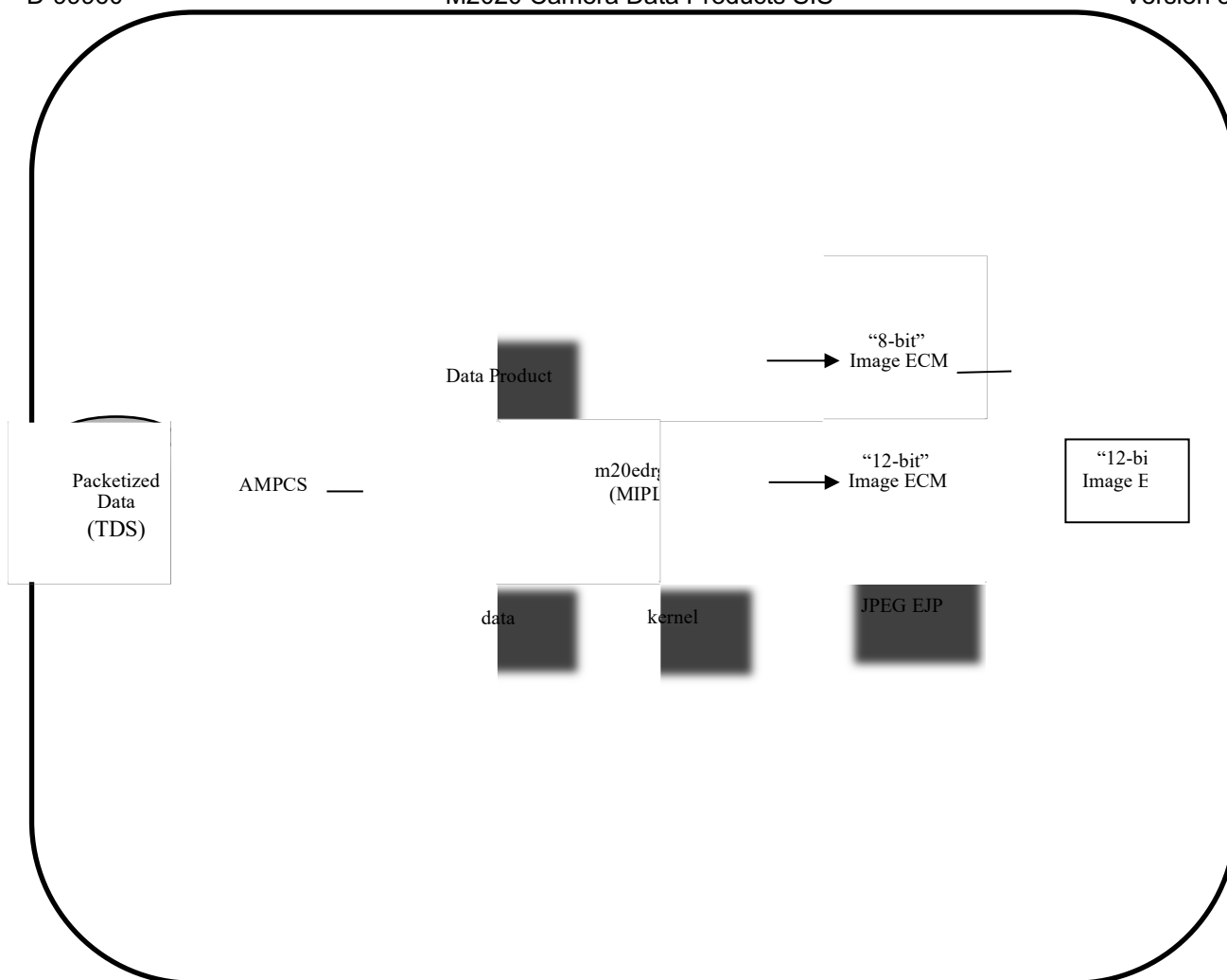
The EDR processing begins with the reconstruction of packetized telemetry data resident on the TDS by the Advanced Multi-mission data Processing and Control Subsystem (AMPCS) into a binary ".dat" data product and associated ".emd" Earth metadata file. The data product and metadata are written by AMPCS to the Operations Cloud Store (OCS) and messages are generated on a Java Message Server (JMS) bus, where they are ingested by MIPL's EDR generation software "m2020edrgen" and processed with SPICE kernels provided by NAIF. The raw EDR will be generated after the JMS notification has been received by the IDS pipeline system. The decompanied and debayered EDRs are then generated by the pipeline. The data flow from instrument to IDS is illustrated in Figure 5-1. The data flow within the IDS pipeline to EDR generation is illustrated in Figure 5-2.



**Figure 5-1: Instrument Data Flow to IDS**

In all EDR cases, missing packets will be identified and reported for retransmission to the ground as “partial datasets”. Prior to retransmission, the missing EDR data will be filled with zeros. The EDR data will be reprocessed only after all “partial datasets” are retransmitted and received on the ground. In these cases, the EDR version will be incremented so as not to overwrite any previous EDR versions.

Descriptions for the various EDR product types are provided in this section. They are broken down into two groupings: a) Image and b) Image Support. Refer also to Table 5-3 for a mapping between the source M2020 instrument and the EDR product type. Within the Image grouping, things are also broken down by sub-category and image type (see Table 5-2).



**Figure 5-2: IDS EDR generation data flow**

## Image EDR Sub-Categories

Image EDRs attempt to reconstruct as closely as possible the data that came off of the imaging detector (CCD or CMOS). This means that products are uncompressed raster images. There are three sub-categories of image EDRs.

### 5.2.1 Raw EDRs

The Raw EDR (type codes: ECM, ECV, ECZ, EDM, ECR) is what prior missions such as MER and MSL called simply “the EDR”. It has been decompressed into a raster, but otherwise remains unchanged from what has been telemetered. Compression generally consists of JPEG or ICER compression, both of which are lossy, or various forms of lossless compression. Lossy compression is a “necessary evil”; it degrades the data to some extent but it allows many times more data to be sent. Experience with prior Mars landed missions shows that the additional images that can be returned via lossy compression more than makes up for the degradation. Note that great pains are taken on the ground to never use lossy compression in the pipeline.

This means that the format of the raw EDR is highly variable. It could be 8 bits if companded, or 10-13 if not. It could be a color image, a grayscale image, or a single-band image encoded with the Bayer pattern. This variability is consistent with it being “raw” data, but is inconvenient to work with, which is why the other categories exist for M2020.

### 5.2.2 Decompanied EDRs

Many of the M2020 image sensors are naturally 10-13 bits in depth, providing a higher dynamic range than the more common 8-bit sensors. In order to save downlink bits, these images are often “companded”, using a lookup table to convert the data to 8 bits. This companding compresses the dynamic range, but does not generally compromise the use of the data.

The decompanied EDR (type codes: EDR, EVD, EZS, ERD) reverses this companding.

Companding can be either linear (i.e. a bit shift) or using a table. Linear companding is equivalent to just dividing the pixels by 2, 4, 8, etc. and then multiplying back again to reverse the process. This makes it little better than an 8-bit sensor, except that the onboard software can look at the actual dynamic range in order to decide how much shifting to do.

Table-based companding allows a curve to be applied. Often this is a square-root table, which preserves bits at the low end (more dynamic range where it is darker), while still preserving the ability to represent bright areas. This takes advantage of the fact that many forms of image noise scale with overall brightness, so more detail can be preserved on the low end where there is less noise. This type of companding is reversed using an inverse lookup table.

Regardless of the type of companding used, the process is lossy: some information is irretrievably lost. Nevertheless, it is critical to do in some cases; for example JPEG only works with 8-bit data.

The decompanied EDR is uniformly the number of bits acquired by the sensor (10-13, or 8 for those sensors that are naturally 8-bit). If the data was not companded, the decompanied EDR is a copy of the raw EDR.

The cameras that support companding are: All ECAM, ZCAM, SuperCam RMI, SHERLOC-WATSON, SHERLOC-ACI. The others (PIXL MCC, LCAM, EDLcams, and both helicopter cameras are natively 8 bits. Note that MEDA Skycam is 12 bits, however, there is no companding process onboard, so it is expected that the pre-decompanded and post-decompanded images will be equal.

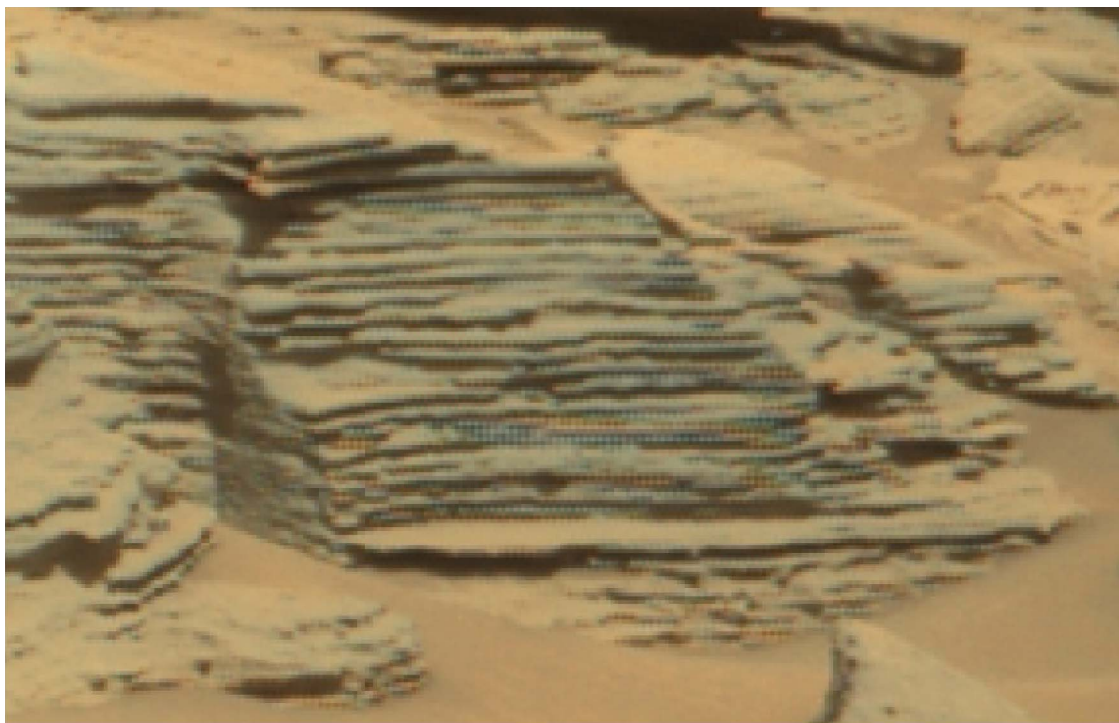
### 5.2.3 Debayered EDRs

If the image is Bayer-encoded, it is debayered using the Malvar algorithm [Malvar2004]. This process is also known as “demosaicking”. This is the same algorithm as is used onboard when images are sent as JPEG.

The Malvar algorithm is one of several de-mosaicking algorithms in widespread use today. Malvar was also used on MSL for the Mastcam, MAHLI, and MARDI instruments (on-board and ground), and for the InSight mission. It is a very simple algorithm to implement (consisting of a set of simple convolution kernels) yet delivers very good results in most cases.

The Malvar algorithm produces three images, one for each color band. The missing color bands at each pixel are created using the assumption that chrominance (color) varies slowly, while luminance (intensity) varies quickly over the scene. The chrominance is determined by examining neighboring pixels; variations from this are assumed to be luminance changes. Thus the reconstructed image retains the full spatial resolution of the raw image. This is accomplished by applying convolution kernels to the raw Bayer-pattern image; see the Malvar paper for details.

Malvar can introduce artifacts into the image in areas where the chrominance varies rapidly. These generally manifest as a “zipper” pattern in the image (See Figure 5-3). Experience with MSL shows that these artifacts do not generally impede operations or science on the images.



**Figure 5-3: An example of a "zipper" pattern: image processing artifact introduced by the Malvar de-mosaicking algorithm (MSL Mastcam M100 Sol 1155)**

Unlike the first two sub-categories, all debayered EDRs have the same type code, EBY. See Table 5-2. Note that if the decompressed EDR is already color, the debayered EDR is not created. This fact, along with the ECAM tiling (Section 5.6.1.1), are the primary motivators for creating the FDR type (Section 6).

The following cameras use the Bayer pattern: all ECAM, ZCAM, SuperCam RMI, and SHERLOC-WATSON. For SuperCam only the JPEG-compressed images go through the de-mosaicking onboard, and not the raw and ICER-compressed images. Additionally, the EDL cameras and Heli-RTE are color cameras, but there is no access to the raw Bayer pattern. The rest of the cameras are all grayscale (single band): PIXL MCC, MEDA Skycam, SHERLOC-ACI, LCAM, and Heli-Nav.

## 5.3 EDR Image Types

Orthogonal to the sub-categories are the image types. See Table 5-2 for a breakdown of how sub-category and image type match to the standard 3-letter type codes.

Note that unlike ECM/EDR, EBY does not distinguish the alternate file types (Z-stack, video, etc).

### 5.3.1 Standard Images

Standard images are just that: simple still-frame images. They include the type codes ECM (raw), EDR (decompanded), and EBY (debayered).

### 5.3.2 Z-stacks (Best Focus Images)

Z-stack images, also called best focus images, are created onboard by combining images taken at different focus settings. The result is a best-focus image with a much greater depth of field. Each pixel contains data from the source image(s) that are in best focus (potentially interpolated between neighboring images). This image may be color or a single band. This EDR type applies to SuperCam, ZCAM, and both SHERLOC cameras.

Z-stack (focus merge) products (created onboard for WATSON, ACI, Mastcam-Z, and RMI) use the type codes ECZ (raw) and EZS (decompanded). Note that Mastcam-Z, WATSON, and RMI Z-stack products are created onboard using de-Bayered images, so the best-focus image product will always be in color (and there will be no EBY product, which is only created for ground de-Bayering). Note that for Mastcam-Z and WATSON, the onboard algorithm does the de-Bayering; the source images must actually be Bayer-pattern raw images. As ACI is a grayscale imaging system (no Bayer pattern microfilters for color), ACI best-focus image products will always be grayscale.

NOTE: The metadata regarding vehicle state for Z-stack images reflects the state of the vehicle at the time the Z-stack product itself was created. This is generally not the same as the vehicle state at the time the source images were acquired. Most importantly, this means the arm state and mast state, and thus camera model, for these images do not properly describe the image. The metadata must be obtained from the thumbnail or full frame of one of the images that went into making the Z-stack. Determining the proper image to use is discussed in 5.3.2.1.

Z-stacks are indicated in the VICAR/ODL label by GROUP\_APPLICABILITY\_FLAG=TRUE in the ZSTACK\_REQUEST\_PARMS group. In the PDS4 label, they are indicated by <img:active\_flag> of "true" in the <img:Focus\_Stack> class.

#### 5.3.2.1 Z-stack parent images

Mastcam-Z, WATSON, and ACI onboard focus merge (a.k.a. z-stack) products are created in the manner described for the MSL MAHLI instrument [Edgett 2012] and [Edgett 2015]. They can be produced from focus stacks consisting of 2 to 8 consecutively acquired images. The onboard merge creates two products, a best-focus image and a range (a.k.a. depth) map. Both products are JPEG-compressed; the range map is always grayscale, the best-focus image is either a color (demosaicked Bayer pattern) or grayscale product, depending upon the input data (i.e., ACI images are always grayscale). Input parent (a.k.a. source) images are stored onboard in NVM (non-volatile or flash memory) within the instrument DEA (digital electronics assembly). Note that ACI and WATSON share a DEA, whereas each of the two Mastcam-Z cameras has its own DEA.

Each Mastcam-Z, WATSON, and ACI onboard focus merge product label reports two vital parameters that relate the product to its parent focus stack images. The first is the START\_IMAGE\_ID field located in the ZSTACK\_REQUEST\_PARMS group. The START\_IMAGE\_ID is an indicator of the camera data product identifier

(CAMERA\_PRODUCT\_ID; a.k.a., IMAGE\_ID) of the first parent image in a given merged focus stack. The CAMERA\_PRODUCT\_ID, equivalent to IMAGE\_ID, is found in the IDENTIFICATION\_DATA\_ELEMENTS group of the parent image's label.

The other parameter is the ZSTACK\_IMAGE\_DEPTH, also reported in the ZSTACK\_REQUEST\_PARMS group in the focus merge product label. This indicates how many images (2 to 8) were merged onboard to create the z-stack product. Knowing the START\_IMAGE\_ID (CAMERA\_PRODUCT\_ID of the first parent image merged), and the ZSTACK\_IMAGE\_DEPTH (the number of consecutively-acquired images merged), identifies all of them. For example, if the ZSTACK\_IMAGE\_DEPTH is 8 and the START\_IMAGE\_ID is 2959, then the parent images are those acquired consecutively with CAMERA\_PRODUCT\_IDs of 2959 through 2966.

It is important to note that onboard focus merges can be performed at any time after a given focus stack was acquired, as long as the parent images remain stored in the DEA. Some merges may be performed on the same sol that the parent images were acquired, but there have been cases (e.g., on MSL MAHLI) in which merges were performed days or even months later.

It is also important to understand that CAMERA\_PRODUCT\_IDs (IMAGE\_IDs) are recycled within the DEA as data are deleted from its NVM storage. Deletions occur at a rate that is dependent upon how much new data is acquired. Usually, deletions occur after several months (sometimes more than a year, if new data are acquired slowly). Because deletions do occur and CAMERA\_PRODUCT\_IDs (IMAGE\_IDs) are re-used, identification of the parent images for a given focus merge product requires finding the most recent usage of that START\_IMAGE\_ID before the merge took place. For example, a focus merge was performed on Perseverance's 90<sup>th</sup> sol to create a best-focus image and range map product for a WATSON target named Hastah\_tsaadah (acquired at a working distance near 25 cm). The merge product CAMERA\_PRODUCT\_IDs (IMAGE\_IDs) are 943 and 944 (representative filenames SI1\_0090\_0674905233\_960ECZ\_N0040048SRLC00003\_000095J01 and SI1\_0090\_0674905237\_960EDM\_N0040048SRLC00003\_000095J01). The START\_IMAGE\_ID reported in the ZSTACK\_REQUEST\_PARMS group for these products is 906. The ZSTACK\_IMAGE\_DEPTH reported is 8. This means that the merge products were created from parent images with CAMERA\_PRODUCT\_IDs of 906 through 913. Reviewing the data acquired before the merge products with IDs 943 and 944 were created shows that the most recent DEA usage of CAMERA\_PRODUCT\_IDs 906–913 occurred on Sol 88, thus identifying the parent images. For WATSON and ACI, the FOCUS\_POSITION\_COUNT (a.k.a. motor count) in the INSTRUMENT\_STATE\_RESULTS group for each of the parent images can then be used to estimate the range and scale (for WATSON and ACI via equations [Bhartia 2021], or future refinements) for each pixel in the range map product which, in turn, provides an estimate of the scale of features in the best-focus image product.

### 5.3.3 Depth Maps

As part of Z-stack processing, the Mastcam-Z and SHERLOC cameras also produce Depth Map products. It is always produced when a best-focus image product is created. This 8-bit, single-band EDR (type code: EDM) applies to SuperCam, ZCAM, and both SHERLOC cameras, and indicates which image the best-focus data came from.

Pixels with a digital number (DN) of 255 indicate the first image in the Z-stack, while a DN of last

This document has been reviewed and determined not to contain export controlled technical data.

image in the stack is dependent on the number of images merged (max 8). Linear scaling is applied between. For example, with a 5-image stack and expected DN's of 51, 102, 153, 204, and 255, a DN of 127 would indicate the depth is halfway between images 3 and 4. Depth maps are not companded and the Bayer pattern does not apply.

**Table 5-1: Relation between commanded image participant in an onboard SHERLOC WATSON,ACI or Mastcam-Z focus merge and depth map grayscale data value (DN) [Edgett2019]**

image commanded to be merged	DN for 8-image merge	DN for 7-image merge	DN for 6-image merge	DN for 5-image merge	DN for 4-image merge	DN for 3-image merge	DN for 2-image merge
1st	255	255	255	255	255	255	255
2nd	223	218	212	204	191	170	127
3rd	191	182	170	153	127	84	—
4th	159	145	127	102	63	—	—
5th	127	109	84	51	—	—	—
6th	95	72	42	—	—	—	—
7th	63	36	—	—	—	—	—
8th	31	—	—	—	—	—	—

As with Z-stack, the metadata regarding vehicle state for Depth Map images reflects the state of the vehicle at the time the Depth Map product itself was created. This is generally not the same as the vehicle state at the time the imagery was acquired. Most importantly, this means the arm state and mast state, and thus camera model, for these images do not properly describe the image. The metadata must be obtained from the thumbnail or full frame of one of the images that went into making the Depth Map. Determining the proper image to use is beyond the scope of this document, but can often be inferred by inspection of the available data.

### 5.3.4 Video Frames

The Video Frame is identical in format to the standard image. It represents a single frame of a video sequence. The EDR type is separate in order to better distinguish video frames from still frames. This EDR type applies to ZCAM and SHERLOC.

This should not be confused with the videos that come out of the EDL cameras (Section 5.9.1). Those cameras directly generate MPEG movies, from which key frames are extracted for archive. The Video Frames described here represent the entire movie from the camera; no combination MPEG or other movie file is made by the flight system.

Video frames include the type codes ECV (raw), EVD (decompanded) and EBY (debayered).

Video frames are indicated in the VICAR/ODL label by GROUP\_APPLICABILITY\_FLAG=TRUE in the VIDEO\_REQUEST\_PARAMS group. In the PDS4 label, they are indicated by <img:active\_flag> of "true" in the <img:Video> class.

### 5.3.5 Recovered Images

A Recovered Image is a ZCAM or SHERLOC image for which the metadata is unavailable. It existed in camera memory but for whatever reason the onboard data product describing it was lost or deleted. Such products have extremely limited metadata.



Most images that are sent to the rover by cameras are stored in Data Products (DPs) onboard, containing both the image and the metadata. These data products are eventually transmitted to Earth, becoming the “.dat” and “.emd” files (Section 5.1). However, the ZCAM and SHERLOC cameras often create what are called “Virtual Data Products” (VDP). These VDPs consist of only the metadata, with a pointer to camera memory to find the actual image. When transmitted, the rover FSW combines them both into a standard DP, after which the VDP is deleted. The images end up looking like any other images. These cameras can also create standard DPs.

Regardless of whether a DP or VDP was created, the image can still remain in the camera memory. Without a VDP to point to it, the rover knows nothing about this image (the camera team eventually deletes them to free up space in the camera). However, the image can still be (re-)sent to Earth. Since the rover knows nothing about the image, it cannot contribute any metadata about the rover state at the time of image acquisition. The only metadata is that available in the camera mini-header (which is only 64 bytes long). Images sent this way are called “recovered” products – they were recovered from camera memory with no additional information.

The most common use case for this is to retransmit an image later at a better compression (often lossless). The initial DP/VDP may have been JPEG, but the recovered product is sent lossless. Recovered products are also commonly used for movies (such as the Mastcam-Z movies of the helicopter) although these tend to be highly compressed.

The minimal metadata has no camera pointing information, and does not even identify the image as a video or still frame. A special product type code is thus used for recovered products, to distinguish them from normal products: ECR for the raw image, and ERD for the decompressed version. No further processing is done in the pipeline on recovered images; specifically, FDRs are not created.

In order to make these images usable, the pipeline attempts to reconstruct the metadata using the appropriate thumbnail image. This thumbnail comes from the original DP/VDP transmission of the image, and should always exist. The idea is to copy the metadata from the thumbnail, and update the parts that come from the mini-header (such as compression type) as well as geometry items such as downsample, subframe, and camera model. For video products, only one thumbnail is made for each GOP (group of pictures), typically 8 frames. This thumbnail is used, with appropriate SCLK adjustment, for all of the video frames in the GOP.

The reconstructed image becomes an ECM (still) or ECV (video) image, and participates in all the normal pipeline computation, including FDR production and stereo processing.

Users are thus advised to ignore the ECR/ERD recovered products, and instead look for the FDR or other type. Recovered products (other than videos) will typically have more than one FDR (and subsequent products) for the different compression levels; these are distinguished in the filename. The filename is designed such that sorting in ascending ASCII order will put the best compression last in the list, so users can easily find the best (least lossy) compression available (see Section 19.3)

### 5.3.6 Raw JPEG EDR (EJP)

There is one additional EDR image type that does not conform to the pattern, but is worth special mention. For images that are sent as JPEG, the EJP type contains the original JPEG stream as downlinked from the spacecraft. It has not been uncompressed or recompressed. This type is not produced unless JPEG compression was used onboard. It may be a color (three band) or grayscale (single band) JPEG.

This document has been reviewed and determined not to contain export controlled technical data.

EJP types may be created for ECAM, SuperCam, ZCAM, SHERLOC, PIXL, EDL, LCAM, and both helicopter cameras.

Note that this is literally the raw JPEG stream from telemetry. In the case of SuperCam RMI, that means that the EJP has *not* been rotated or reflected. Its geometry will thus not match the rest of the EDRs. Also note that EJP type may not exist for thumbnails, even if they're JPEG.

EJPs are not used for processing in any way. They are included in the archive in case useful information about compression noise can be extracted from the compressed stream. They do not get their own PDS label, but rather are included as a supplemental product in the raw EDR image label.

## 5.4 EDR Image Sampling

There are a few generic types of image sampling that apply to most cameras. These are not separate products (with the exception of thumbnails) but are rather ways that the image may have been processed onboard.

### 5.4.1 Subframing

Subframing is the process of cutting out a subset of rows and columns of the full-frame image. It can be thought of as creating a window on the detector, with the same resolution but smaller coverage area. This is often done to reduce downlink bandwidth, when there is only a small region of interest. Note however that almost all ZCAM images are subframed, in order to remove the non-photosensitive pixels from the edges of the sensor.

Note that ECAM Tiling (see Section 5.6.1.1) is an application of subframing, although tiles themselves can also be subframed.

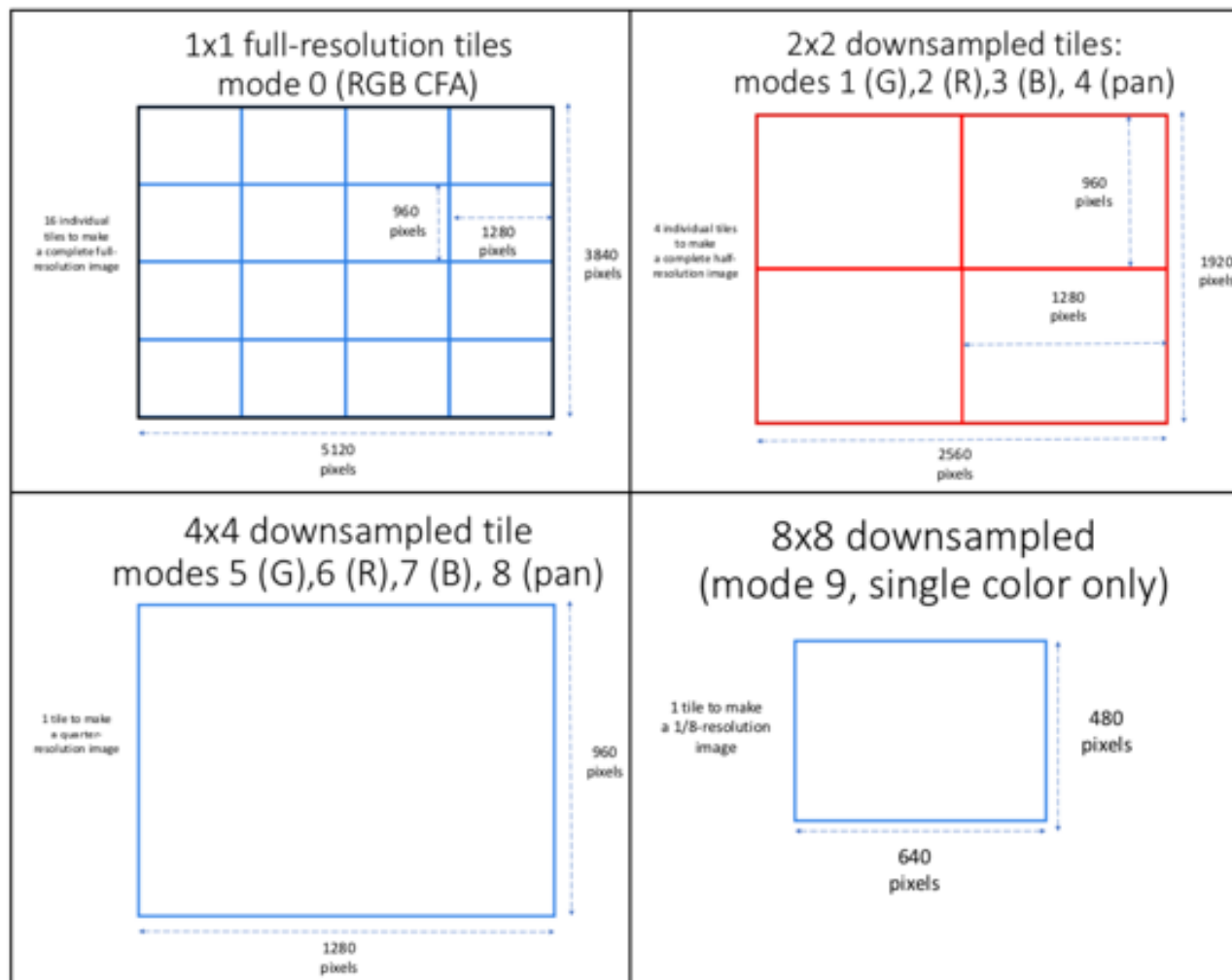
### 5.4.2 Downsampling

Downsampling creates a smaller version of the image, resulting in reduced resolution of the same coverage area. Downsampling can be done via one of three methods: 1) nearest neighbor pixel averaging, 2) pixel averaging with outlier rejection, or 3) computing the median pixel value.

Note that downsampling can be applied to subframes. In this case, the subframe is measured with respect to the full sensor, before downsampling is applied.

### 5.4.3 Pixel Binning

Pixel binning is a process whereby information from the focal plane is downsampled by grouping and averaging pixels. The Engineering Cameras use pixel binning when downsampling from the FPGA (see Figure 5-4). The image maybe further downsampled via software, as described in Section 5.4.2. Color band specific binning is supported for 2 x 2 ("modes 1 - 3") and 4 x 4 ("modes 5 - 7") binning only. "Mode 9" will produce an extra low resolution image (8 x 8 binned) from all color bands similar to "mode 4" and "mode 8". The area covered by a non-binned image is 1/16 of the entire focal plane (and field of view). The area covered by 2 x 2 binned images is 1/4 of the entire focal plane (and field of view). The area covered by 4 x 4 binned images is the entire focal plane (and entire field of view) with resolution similar to images from MER and MSL engineering cameras. An 8 x 8 binned image downsamples the entire focal plane (and field of view) to produce a tile that is 1/4 standard size (image and data).



**Figure 5-4: ECAM multiple modes of pixel binning.**

#### 5.4.4 Co-adding

Co-adding increases the signal-to-noise ratio, resulting in a higher quality image. The Engineering Cameras are the only instruments that have this capability. The co-adding function begins by acquiring an image from the focal plane to the accumulator buffer and then acquiring an image of the same scene from an additional exposure to the scratch buffer and then adding it to the accumulator buffer to create a new "image". This process is repeated, as commanded by the user, until the final image has been added and then the total summed image that resides on the accumulator buffer is divided by the number of co-added images before transmission to FSW. SNR improves in a co-added image by the square root of the number of co-added images. For ease of implementation purposes, co-adds must be a power of two and a maximum of 16, resulting in only five co-adding options: 1 (equivalent to no co-adding), 2, 4, 8 or 16. A co-added image contains exactly the same amount of data as a non-co-added image (1280 x 960 bits, 12 bits per pixel = 14,745,600 bits). The default setting for co-adding on the EECAM FPGA is 1.

#### 5.4.5 Thumbnails

Thumbnails are a reduced-resolution version of the original image, sent in addition to, or instead of, the original image. They apply to all image EDRs. The main purpose of a Thumbnail EDR is to provide an image summary using a very low data volume compared to the original image. Decisions

about downlinking the original image can be made using the Thumbnail. For the SuperCam RMI, Thumbnails are produced relative to the full-frame image, even if the product is downsampled or subframed. For Mastcam-Z and SHERLOC camera images, thumbnails are of the subframe region only, and not the full-frame. For ECAM, the thumbnail covers whatever was actually read from the camera, before any additional processing in the FSW. This generally means the tile, but there are corner cases where there might be differences.

## 5.5 EDR Product Format

Description of EDR product formats in this section will be by instrument suite.

The EDR will be formatted according to this SIS, following the general terms of labeling and bit ordering previously discussed in Sections 3 and 4.2, respectively. The various EDR formats and their data sizes, across all instrument suites, are listed in Table 5-2 and Table 5-3 and are discussed subsequently in this section.

**Table 5-2 – Type Codes for Image EDRs**

	Raw	Decompanded	Debayered
Normal Image	ECM	EDR	EBY
Video Frame	ECV	EVD	EBY
Z-Stack	ECZ	EZS	EBY
Depth Map	EDM	n/a	n/a
Recovered Product	ECR	ERD	n/a

**Table 5-3 – List of EDR Types and Formats**

Type	Product Identifier	Format (bits)	Instruments (abbr.)	Description
Original Image Product	ECM	8-bit or 16-bit unsigned	All cameras	Contains companded data either as originally downlinked or JPEG decompressed. It may be a color or grayscale.
JPEG	EJP	8-bit unsigned	ECAM, ZCAM, WATSON, ACI, MCC, EDL, HELI	Contains JPEG-compressed data as originally downlinked. It may be a color or grayscale (single band) JPEG.
Video frame	ECV	8-bit or 16-bit unsigned	ZCAM, WATSON, ACI, EDLcam	Identical in format to the Original Image Product. It represents a single frame of a video sequence.
Z-stack	ECZ	8-bit or 16-bit unsigned	ZCAM, WATSON, ACI, RMI	A best-focus Z-stack image, created onboard the instrument from a combination of images at different focus settings, with a much greater depth of field. Typically SHERLOC-WATSON data. It may be color or a single band.
Depth map	EDM	8-bit unsigned	ZCAM, WATSON, ACI	Second of two products produced by onboard focus merge (z-stacking). This single-band grayscale image provides information about which images the best-focus data came from, and can provide a crude measure of target surface relief.  This single-band image indicates which image the best-focus data came from. DN of 255 indicates first image in the Z-stack, while the last is dependent on number of image-merge[See Table 5-1], with linear scaling between.
Recovered	ECR	8-bit or 16-bit	ZCAM,	A Product for which the metadata is unavailable.

This document has been reviewed and determined not to contain export controlled technical data.

Type	Product Identifier	Format (bits)	Instruments (abbr.)	Description
Product		unsigned	WATSON, ACI	
Thumbnail	ALL	8-bit or 16-bit unsigned	ECAM, RMI, ZCAM, WATSON, ACI, MCC, SkyCam, EDLcam, HELI, LCAM	This data product is a spatially sized down version of an existing Full Frame, so is less than full size and less than full resolution. The bit scaling rules described for the Full Frame case above also apply here.
Original Image Product (De-companded)	EDR	8-bit or 16-bit unsigned	ECAM, RMI, ZCAM, WATSON, ACI, MCC, Skycam, EDLcam, HELI, LCAM	Nominal data product. If "12 to 8-bit" scaling was performed, the 12-bit data has been unscaled back to 12 bits, stored as the last 12 bits of a 16-bit integer.
Video frame (De-companded)	EVD	16-bit signed integer	ZCAM, WATSON, ACI, EDLcam	Identical in format to the Original Image Product. It represents a single frame of a video sequence.
Z-stack (De-companded)	EZS	16-bit signed integer	ZCAM, WATSON, ACI, RMI	A best-focus Z-stack image, from a combination of images at different focus settings, with a much greater depth of field. Typically SHERLOC-WATSON data. It may be color or a single band.
Recovered Product (De-companded)	ERD	16-bit signed integer	ZCAM, WATSON, ACI	A Product for which the metadata is unavailable.
De-bayered	EBY	16-bit signed integer	ECAM, RMI, ZCAM, WATSON, EDLcam	Contains de-bayered color image.
Row Summed	ERS	32-bit unsigned	ECAM, SkyCam	nx1 array of 32-bit integers whose length is equal to image height, wherein the DN value for the Jth element equals the sum of all pixels in the Jth row.
Column Summed	ECS	32-bit unsigned	ECAM	1xn array of 32-bit integers whose length is equal to image width, wherein the DN value for the Jth element equals the sum of all pixels in the Jth column.
Reference Pixel	ERP	16-bit unsigned	SkyCam	Dark pixels bookending (pre- and post-) image pixels during serial register readout. There are "pre-" Reference and "post-" Reference pixels.
Histogram	EHG	32-bit unsigned	ECAM	DN histogram computed from image can have either 256 or 4096 bins, each capable of holding count values of up to 4,194,304.
IDPH only	EID	8-bit unsigned or 16-bit signed integer	ECAM	IDPH (Image Data Product Header) only, with no image data. The data is formatted as a 1x1 image with a 0 pixel value.
NVM Status RAM Status	E48 E49	32-bit unsigned integer	PIXL	Non-volatile memory or Random Access Memory system parameter dump.
TRN	ESO	CSV	PIXL	Translation Relative Navigation(TRN) data from the MCC
SLI Estimate	ESF	CSV	PIXL	The distance and plane solutions derived from SLI measurements.
Centroid Image	ESA	32-bit unsigned integer	PIXL	Estimated circle center and radius of the centroid

Type	Product Identifier	Format (bits)	Instruments (abbr.)	Description
Circle Centroid	ESC	CSV	PIXL	The centroid positions (X,Y) and intensity of centroid on the CCD.
Image ROI	ESR	32-bit unsigned integer	PIXL	Region around each centroid in uncompressed format. The ROI of centroids are coded sequentially.
State of Health	SOH	Variable depending on the data type of the engineering value	RMI	Comprised of binary metadata describing the health and safety of SuperCam RMI
Debug Info	ED1		SRLC	Pass-through
Directory Dump	ED2		SRLC	Pass-through
Image Status	ED3		SRLC	Pass-through
Zstack List	ED4		SRLC	Pass-through
Memory Dump	ED6		SRLC	Pass-through
BIST	ED7		SRLC	Pass-through
Util-Test	EUT		SRLC	Pass-through
Movie	ECV	mpeg	EDL	MPEG movie files
Movie Frame	ECV	8	EDL	Single frame extracted from MPEG file
Microphone	EAU	8	EDL	Audio files
Navigation Map	TBD	TBD	TBD	Navigation maps come from onboard autonav processes. They are not currently being processed by IDS, but may in the future.

## 5.6 Instrument-Specific Image EDR Details

### 5.6.1 Engineering Camera Instrument Suite EDRs

Although the ECAM detectors and on-camera memory allow acquisition of images 5120 x 3840 pixels in size, hardware readout limitations require reading out “tiles” from the detector. This results in the full-frame EDR with a maximum dimension of 1296 samples by 976 lines. These tiles are reassembled in the FDR product (Section 6.2.1).

The data from the Engineering Cameras can come down in one of five different ways: color or grayscale JPEG, ICER compressed, losslessly-compressed (LOCO), or uncompressed. These methods apply across all of the image EDR types.

#### 5.6.1.1 ECAM Tiling

The engineering cameras for Mars 2020 are a significant upgrade compared to the MSL cameras. They are 5120 x 3840 pixels (vs. 1024 x 1024 on MSL), and have color Bayer-pattern filters.

However, due to limitations in the MSL-heritage flight software (FSW), images can be no larger than 1280 x 960 (actually 1296 x 976 to accommodate overlap). This means that images must be “tiled”.

Tiling means that only a portion of the image is read in at a time. The entire sensor plane is exposed at once and the results are stored onboard the camera. Separate image reads then pull out the pieces, or “tiles”, of interest. It takes 16 tiles to read out the full-resolution image, or 4 tiles at 2x downsampling. At 4x or 8x, the entire image can be read at once. Only one image can be stored on the camera at a time, so once the next exposure is performed, no more tiles can be read from the previous one.

In the surface FSW, used for most of the mission, this tiling is very flexible. Any arbitrary location can be read from the sensor, at any resolution and any color, subject to the size limitations. Ground software reassembles these tiles (see Section 6.2.1) into the full image.

However, the cruise/EDL FSW is much more limited. This controls the vehicle through the critical EDL phase and the first few days after landing. In order to minimize changes to the cruise FSW from the MSL baseline, a mode was implemented to use “DC offset” to control which tile to read.

In general, DC offset specifies the analog value that is subtracted from the video signal prior to the analog/digital conversion. However, the cruise FSW co-opts this to specify which tile to read out. Note that only one tile can be read from any given exposure; thus building up a full-size, full-resolution image would require 16 separate exposures. The DC offset value corresponds to a lookup table that defines a specific readout mode and portion on the detector. The table has a range of 0-35 as defined below. Each tile is 1280 samples by 960 lines (with no overlap).

**Table 5-4 – DC Offset specification for the Engineering Cameras**

DC Offset	FPGA Mode	Bayer Channel	Binning	Start Row	Start Sample
0	0	Bayer	None	0	0
1	0	Bayer	None	0	1280
2	0	Bayer	None	0	2560
3	0	Bayer	None	0	3840
4	0	Bayer	None	960	0
5	0	Bayer	None	960	1280
6	0	Bayer	None	960	2560
7	0	Bayer	None	960	3840
8	0	Bayer	None	1920	0
9	0	Bayer	None	1920	1280
10	0	Bayer	None	1920	2560
11	0	Bayer	None	1920	3840
12	0	Bayer	None	2880	0
13	0	Bayer	None	2880	1280
14	0	Bayer	None	2880	2560
15	0	Bayer	None	2880	3840
16	2	Red	2x2	0	0
17	2	Red	2x2	0	2560
18	2	Red	2x2	1920	0
19	2	Red	2x2	1920	2560
20	1	Green	2x2	0	0
21	1	Green	2x2	0	2560
22	1	Green	2x2	1920	0
23	1	Green	2x2	1920	2560
24	3	Blue	2x2	0	0
25	3	Blue	2x2	0	2560
26	3	Blue	2x2	1920	0
27	3	Blue	2x2	1920	2560
28	4	Panchro	2x2	0	0
29	4	Panchro	2x2	0	2560
30	4	Panchro	2x2	1920	0
31	4	Panchro	2x2	1920	2560
32	5	Red	4x4	0	0
33	6	Green	4x4	0	0
34	7	Blue	4x4	0	0
35	8	Panchro	4x4	0	0

### 5.6.1.2 CacheCam

The CacheCam views the sample through a mirror. In order to compensate for this, the EDR generation process flips the image vertically. No rotation is performed, because the rotation angle of the sample tubes is arbitrary. Proper scientific interpretation of the samples is facilitated by them being flipped correctly, so they look as they would when the tube is opened on Earth. See Figure 5-5, which shows a capped sample tube from sol 196.

Care must be taken when commanding CacheCam subframes, as the commands work with the original, unflipped coordinates.

The flip is taken into account in the label: the subframe (FIRST\_LINE in the IMAGE\_DATA and SUBFRAME\_REQUEST\_PARM groups) has been adjusted to account for the flip (thus making the flip transparent to most users of the data). However, the DETECTOR\_FIRST\_LINE (in the INSTRUMENT\_STATE\_PARS group) has not been adjusted for the flip; the DETECTOR\* keywords represent (per the definition) the actual coordinates on the detector.





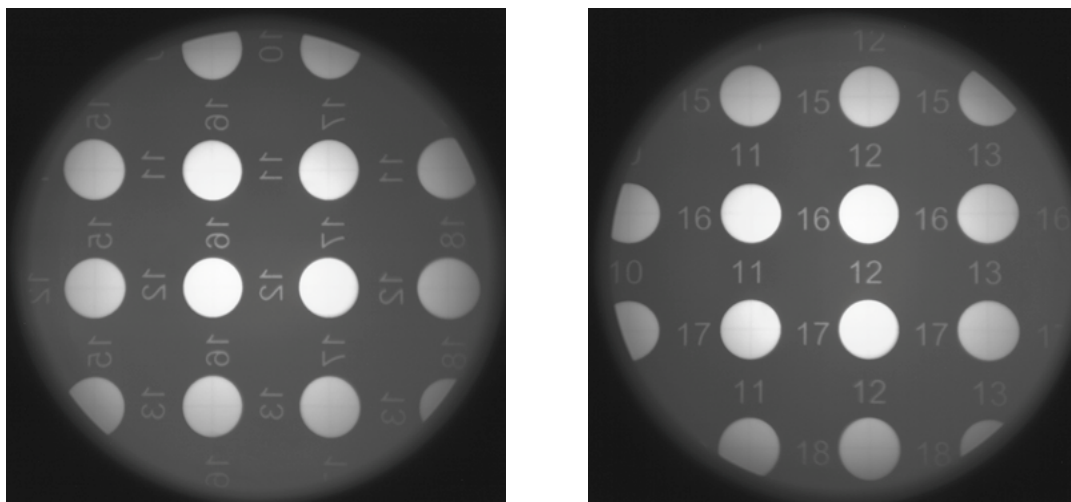
**Figure 5-5: CacheCam rotation. Left: original EJP. Right: raw EDR after flip**

### 5.6.2 SuperCam RMI EDRs

Depending on the size of the image, the RMI science data could require up to 3 transfers from the instrument image buffer to the flight system. The flight system will concatenate the data together into one data product to be downlinked to Earth. The image data, telemetered on a variable number of bits (up to 13-bits uncompressed), will be decoded in single frame form and stored in the lowest 10 to 13-bits of a 16-bit integer as the image EDR. Data returned as compressed are ICER (color) or JPEG (color) encoded and will be decompressed as part of the EDR processing.

The optics of SuperCam introduce a 90 degree rotation and a mirror image to the raw data. In order to make interpretation of the results easier, the raw EDR generation process flips and rotates the image so it is right side up compared with images taken by the other mast-mounted cameras (Navcam, Mastcam-Z). Care must be taken when commanding the RMI to account for this rotation. The EJP, being the raw JPEG bit stream (when JPEG is used) is neither rotated nor flipped. (See Figure 5-7).

See Section 3.6 for a general description of the FITS EDR format for SuperCAM data.



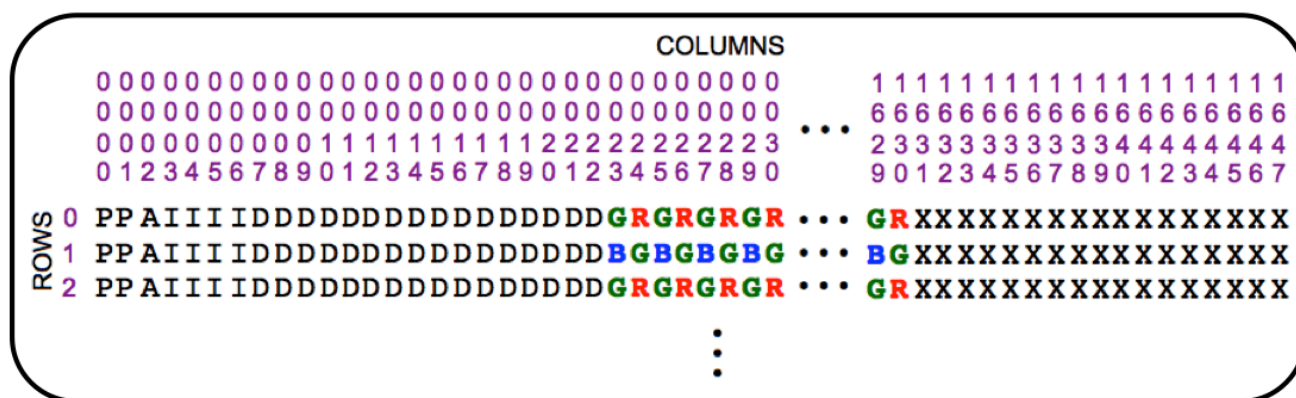
**Figure 5-6: RMI Rotation. Left: original EJP. Right: raw EDR after rotation and flip.**

### 5.6.3 Mastcam-Z (ZCAM) EDRs

The Mastcam-Z cameras create images of up to 1648x1200 pixels. Of the 1648 pixels per line, only 1608 are photoactive pixels. The line structure is shown in Figure 5-7 below, and is broken down into the following:

- 2 dark pixels from the end of the previous line (“P” in Figure 4.3.3.1)
- 1 invalid ADC pipeline pixel from the interline time (“A” in Figure 4.3.3.1)
- 4 isolation pixels (“I” in Figure 4.3.3.1)
- 16 dark pixels (“D” in Figure 4.3.3.1)
- 1608 photoactive Bayer-pattern RGB pixels (“R”, “G”, “B” in Figure 4.3.3.1)
- 17 dark, isolation, and overscan pixels (“X” in Figure 4.3.3.1)

Because there are an odd number of starting dark pixels, the first valid pixel (23) on even lines will be G (GRGRGR...) while on odd lines it will be B (BGBGBG...).



**Figure 5-7: ZCAM – RGB in Bayer Pattern Layout on CCD**

The Mastcam-Z is equipped with 6 narrow-band geology filters, a wide-band color filter, and a solar filter in each eye. These are listed in Table 2-8. Because the Bayer filters cannot be turned off, they must be accounted for when using the geology filters. For the visible-light filters, we simply use the Bayer color band closest to the geology filter, with bilinear interpolation. However, in the infrared, the passband of all three Bayer filters is nearly identical. This means the Bayer filters are basically transparent in the infrared, and can be ignored. Thus, the image can be used as-is without any debayering. This is summarized in Table 5-5, where “Malvar” means normal color interpolation, “Identity” means to pass through unchanged, and “Red”, “Green”, and “Blue” mean bilinear interpolation of those bands only. Note that the same processing is done for both JPEG and non-JPEG images; for JPEG it is done onboard by the camera, while for non-JPEG it is done by the IDS pipeline (EBY product).

**Table 5-5 – Bayer pattern interpolation per filter for Mastcam-Z**

Camera	Filter 0	Filter 1	Filter 2	Filter 3	Filter 4	Filter 5	Filter 6	Filter 7
Left	Malvar	Red	Red	Red	Red	Green	Blue	Malvar
Right	Malvar	Red	Identity	Identity	Identity	Identity	Identity	Identity

## 5.6.4 SHERLOC Camera Instrument Suite EDRs

The data from the SHERLOC cameras can come down in one of four different ways: color or grayscale JPEG, losslessly-compressed, or uncompressed. These methods apply across all of the image EDR types (standard image, Z-stack, depth map, video frame). Note that ZCAM, WATSON, and ACI onboard focus merge products will always be JPEG-compressed.

### 5.6.4.1 WATSON

The SHERLOC-WATSON camera creates images similar to Mastcam-Z, including the sample layout shown in Figure 5-7. It has no filter wheel. However, it does have LEDs that can be used to illuminate the scene in white light and 365 nm UV light. It also has a transparent cover that is normally open for imaging operations. If it is closed and the cover is clean, it imparts a slight dimming to the scene. If the cover has a film of dust on it (which is likely, based on MSL EDL experience), it imparts a slight color change. See the WATSON documentation.

### 5.6.4.2 ACI

The SHERLOC-ACI camera creates images similar to WATSON, but there is no CFA (Bayer pattern) on the detector and thus all images will be grayscale. ACI also has the same LED suite as WATSON (2 groups of white LEDs and 1 group of 365-nm UV LEDs).

## 5.6.5 PIXL MCC EDRs

The image data from the PIXL MCC comes down in one of two different ways: raw (uncompressed) or JPEG compressed. The PIXL imager is grayscale so there is no Bayer pattern.

The MCC is mounted at an approximately 18° angle to the boresight of the X-ray. Since in normal operation the X-ray is oriented normal to the local surface, this means the MCC EDR is skewed with respect to the surface, creating a trapezoidal distortion. This effect is removed in RDR processing, see Section 14.3.

## 5.6.6 MEDA Skycam

The data from the Skycam come down in one of two different ways: uncompressed or compressed. Data returned as compressed are ICER or LOCO encoded and will be decompressed as part of EDR processing.

Note that `scik` both in the filename and populated in **IDENTIFICATION** data elements (see 3.7) comes from the Timetag field in the telemetry. Because Timetag field does not have subseconds, the `msec` field in the filename will always be 0.

## 5.6.7 EDL Camera Suite

The imaging data is stored in the EDLcam as a set of raw, uncompressed frames. These frames can then be sent in any of three ways: color JPEG, compressed MPEG, or raw uncompressed. For each MPEG file, imaging products are generated by converting every frame into an EDR image product. The frame extraction is performed using the FFMpeg framework.

The EDL images were sent in several different ways.

First, a few frames were sent as JPEG right after landing. These are:

```
ESF_0000_0666952928_995ECM_N0000000EDLC00000_000000J01.IMG
```

```
EDF_0000_0666952909_467ECM_N0000000EDLC00002_000000J01.IMG  
EDF_0000_0666952910_467ECM_N0000000EDLC00002_000000J01.IMG  
EDF_0000_0666952913_703ECM_N0000000EDLC00001_000000J01.IMG  
EDF_0000_0666952914_703ECM_N0000000EDLC00001_000000J01.IMG
```

Second, the complete EDL thumbnail movies were sent as a set of MPEG files, on sol 0. Every frame was extracted from these movies, to create the thumbnails (\*ECV\_T\*FH\* in the filename). The FH compression code means “From H.264” (H.264 is the MPEG codec used).

Third, the complete EDL non-thumbnail movies were sent as a set of MPEG files, on sols 1-4. Every frame was extracted from these movies, to create the full-frames (\*ECV\_N\*FH\* in the filename). These frames represent the most complete movie sequences. They may be shorter duration than the thumbnail movies, as the thumbs were used to narrow down the range of the full-frame movies, to discard parts before or after the EDL sequence when nothing was happening.

The PUC1 and RDC full-frame movies were sent in several separate movie files: 3 for PUC1 and 7 for RDC. Due to a commanding issue when building the MPEG files, one frame is missing at the end of each of the sets except the last PUC1. The frame numbers, which are in IMAGE\_ID in the label, are: 2311, 3669 for PUC1, and: 1471, 2154, 2837, 3520, 4203, 4886, 5570 for RDC. All of these frames have been retransmitted losslessly. But in order to make a complete movie from the FH frames, these images must be inserted. They correspond to filenames (using FDR names):

```
EAF_0000_0666952776_704FDR_N0000000EDLC00016_0000LUJ01.IMG  
EAF_0000_0666952821_999FDR_N0000000EDLC00016_0000LUJ01.IMG  
  
EDF_0000_0666952802_102FDR_N0000000EDLC00016_0000LUJ01.IMG  
EDF_0000_0666952824_949FDR_N0000000EDLC00016_0000LUJ01.IMG  
EDF_0000_0666952847_763FDR_N0000000EDLC00016_0000LUJ01.IMG  
EDF_0000_0666952871_110FDR_N0000000EDLC00016_0000LUJ01.IMG  
EDF_0000_0666952893_957FDR_N0000000EDLC00016_0000LUJ01.IMG  
EDF_0000_0666952916_738FDR_N0000000EDLC00016_0000LUJ01.IMG  
EDF_0000_0666952939_624FDR_N0000000EDLC00016_0000LUJ01.IMG
```

Note that the last RDC frame was actually transmitted losslessly twice; the first is listed.

Finally, individual uncompressed, raw frames were sent throughout the mission, as downlink space allows. These are lossless frames that represent the best quality EDLcam images. As of this writing (PDS release 3), most of the “action” frames have been sent in this way, but there are a number of frames still to go. They will come out over time, in future PDS releases, as downlink allows. These frames are identified as lossless using the LU compression code (\*ECM\_N\*LU\* in the filename).

All of the EDL frames end up in sol 00000 for PDS. For ops, the frames are in the sol of downlink as of this writing, but will be moved to sol 00000 at some point.

For each set of movies (separately), order them by SCLK in the filename to get the proper sequence for movie reconstruction. Alternatively, the IMAGE\_ID in the label can be used. Note that the COMMAND\_DISPATCH\_SCLK in the label is the time at which the image was queued for downlink, not the acquisition time. SPACECRAFT\_CLOCK\_START\_COUNT is the actual time of acquisition (matches the filename, but to more precision).

Separately from the EDL movies, the RDC and RUC can be used during normal rover operations, outside of EDL. These images are on the sols when they were acquired (see for example sols 121 and 152).

The ECM/ECV files are converted to EDR/EVD type. This should represent a decompressing step, but the EDLcam data are all in byte space, so the EDR/EVD is simply a copy of the corresponding EDM/ECV. The lossless frames are Bayer-encoded, so they create a EBY type. It is recommended that the FDRs be looked at for consistency rather than the various EDR types. Those patterns would be: \*FDR\_N\*00J01 for JPEG frames, \*FDR\_T\*FH\* for thumbnail extracted movies, \*FDR\_N\*FH\* for full-frame extracted movies, and \*FDR\_N\*LU\* for uncompressed frames.

### 5.6.8 LCAM

Unlike the other cameras, LCAM images are pre-processed by the LCAM team before coming to IDS. This greatly limits the amount of metadata included with the image. Images are presented to IDS in JPEG or PNG format (PNG is lossless), and are decompressed to the standard EDR formats. As it turns out, lossless versions of all the frames were available, thus the EDRs have filenames of the form EL\*ECM\_N\*LU\* .

The LCAM EDR frames do have a camera model, which represents the on-board estimate of the rover's trajectory during descent. However, the FDR frames have a significantly better camera model (see Section 6.3.2).

The ECM files are converted to EDR type. This should represent a decompressing step, but the LCAM data are all in byte space, so the EDR is simply a copy of the corresponding EDM. It is recommended that the FDRs be looked at for consistency rather than the various EDR types.

### 5.6.9 Helicopter

The data from the helicopter cameras are read into onboard helicopter memory. After the flight is over, the data are transmitted to the rover for relay to Earth. As with the LCAM, images are pre-processed by the Helicopter team before coming to IDS, thus limiting the amount of metadata included with the image. Images may be compressed using JPEG or TIFF (TIFF is lossless). There is no facility to retrieve the raw Bayer pattern from the RTE camera.

The RTE camera is 4224x3120 pixels, but only the first 4208 pixels are active. There is no data in the last 16 columns (4209-4224).

#### 5.6.9.1 Helicopter Idiosyncracies

The helicopter was designed as a demonstration mission, with a maximum of 5 flights to demonstrate controlled flight on Mars. As such, minimal resources were provided during development to the creation of image products. After those flights were successful, the helicopter transitioned into a scout mission, with an indefinite number of flights. This lack of early resources led to some compromises in design and implementation that are noted here. There were additional compromises on the flight side; for example, the RTE camera telemetry does not include helicopter state information, which must be inferred on the ground by the helicopter team. None of these inhibit use of the data.

- The RMC is called “Rover” motion counter, as described in Section 4.4.
- The filename should contain the first two elements of the RMC (“Flight” and “Pos” for the helicopter). It does contain the Flight number; however the Pos counter is always 1 in the filename. See Section 4.4.2.2.
- The Pos counter counts Nav and RTE images separately, rather than being a unified counter for both (Section 4.4.2.2).
- The initial version of Heli Nav images contains only the onboard estimate of helicopter pose during flight. These are replaced later by a version containing the helicopter pose as reconstructed by the helicopter team from GNC data. The data delivered to PDS should contain only the latter for flight images, but ground-based images between flights may not be updated. This is not a particular concern though, since the heli pose is by definition identity with respect to the HELI\_G frame on the ground, so there is nothing to update for ground frames. This delay in GNC availability is an issue to be aware of in operations, however.
- The initial version of RTE images contain **no** pose estimate whatsoever. The flight system does not integrate the RTE images with the pose telemetry. As with Nav, a version is created later containing the reconstructed GNC-based pose. Again, PDS flight data should contain only the reconstructed data, but ground images and early images in operations may not.
- Ground images taken between flights have coordinate information (specifically, the HELI\_G to SITE frame transform) in the label for the *next* flight. However, the Flight counter (in both the RMC and filename) indicates the *previous* flight. The position of the helicopter is thus described properly in the HELI\_G frame, it is just mislabeled as to the flight index. The HELI\_M frame definition is correct for both Nav and RTE, since the landed pose in HELI\_M frame is by definition constant with respect to HELI\_G.

## 5.7 Image Support EDRs

The following EDRs are not images, but are related to images and provide additional information about them.

### 5.7.1 Row Summation EDR (ERS)

A row summation EDR is the sum of the rows of a full-frame or subframed image, performed by the FSW. The EDR is a  $n \times 1$  array of 32-bit integers (whose length is equal to the image height) where the DN value of the  $i$ 'th element is the value of the sum of all the pixels in the  $i$ 'th row. Applicable to the Engineering Camera instrument suite and MEDA SkyCam.

### 5.7.2 Column Summation EDR (ECS)

A column summation EDR is the sum of the columns of a full-frame or sub-framed image, performed by the FSW. The EDR is a  $1 \times n$  array of 32-bit integers (whose length is equal to the image width) where the DN value of the  $i$ 'th element is the value of the sum of all the pixels in the  $i$ 'th column. Applicable to the Engineering Camera instrument suite.

### 5.7.3 Reference Pixel EDR (ERP)

The SkyCam onboard detector array has "Reference" dark pixels located before and after the image data. Before the image data in each row there are 1 invalid pixel and 16 reference pixels. Each image row is followed by 14 reference pixels and the camera I.D. The ERP product contains only those portions of the detector. Applicable to the SkyCam only.

### 5.7.4 Histogram EDR (EHG)

The histogram EDR is a 32-bit integer array storing the histogram of the image, performed by the FSW. A 1x256 or 1x4096 array will be returned. Applicable to the Engineering Camera instrument suite.

### 5.7.5 IDPH Only (EID)

IDPH (Image Data Product Header) metadata only, with no image data. The data is formatted as a 1x1 image with a 0 pixel value. This product is generated when the cameras are commanded to acquire a picture (for example, to pre-point the RSM), but no image data are requested from the camera.

## 5.8 Engineering EDRs

These EDR types supplement the information available for various instruments. For more information, see the instrument-specific SIS for each instrument.

### 5.8.1 SuperCam RMI

Note that SuperCam has a microphone, but that is delivered as part of the non-image SuperCam bundle [SIS\_SCAM], and thus is not discussed here.

#### 5.8.1.1 State of Health EDR (SOH)

All SuperCAM FITS EDRs contain a number of SOH as binary tables. The ODL/VICAR label produced during the call to edrgen also contain a set of instrument/rover parameters.

### 5.8.2 PIXL-MCC

All PIXL MCC Engineering EDRs start with 34-byte image header taken directly from the first TM packet from the MCC, followed by variable size MCC specific image data products and ended with Meta Data(if any) as provided by MCC.

The products listed below are the ones that are relevant to the MCC camera (there are many other non-image products, see the non-image bundle [SIS\_PIXL]). The ESO and ESF products are delivered to PDS, but as part of the PIXL non-image bundle and are not discussed further here.

#### 5.8.2.1 Non-Volatile Memory Telemetry Status Report EDR (E48)

The NVM telemetry status report EDR contains the PIXL-MCC system parameters as read from the NVM copy of the parameters in the MCC System memory.

#### 5.8.2.2 RAM Telemetry Status Report EDR (E49)

The RAM telemetry status report EDR contains the PIXL-MCC System Parameters as read from the RAM (volatile) copy of the parameters in the MCC System memory.

#### 5.8.2.3 Translation Relative Navigation EDR (ESO)

This EDR contains the Translation Relative Navigation (TRN) measurement from the MCC. This is a autonomous terrain relative navigation algorithm that operates by extracting and matching keypoints/features between successive images, from which a relative translation is determined. The result can either be output in pixels or micro meters. It is stored in CSV format.

#### 5.8.2.4 Structured Light Illuminator Estimate EDR (ESF)

The SLI algorithm that acquires distances to the observed terrain, using the SLI pattern. The SLI structure contains the distance and plane solution derived from the SLI measurements. It consists of a 26 byte SLI header followed by Nx24 bytes data that provides the individual image and spatial coordinates as well as an SLI beam identifier and goodness of the match for each SLI spot found.

#### 5.8.2.5 Centroid Compression EDR (ESC)

This EDR contains the Centroid compressed image from the MCC in CSV format.

The centroids are listed sequentially. The centroids compression returns only the centroid positions on the CCD. One centroid is 10 bytes long. The bytes 0-3 are X position, bytes 4-7 are Y position both coded as 4 byte unsigned longs. In order to convert to image coordinates, these integers must be divided by 5000000. Bytes 8-9 are the intensity of the centroid with valid data range of 0-320000.

#### 5.8.2.6 Circle Centroid Compression EDR (ESA)

This data product contains the Circle Centroiding (CC) solution as the estimated circle center, radius, and relevant information.

The Circle Centroid (CC) compression executes the autonomous Circle Centroiding algorithm that acquires absolute navigation information, by identifying the location of the circular abraded area. The output telemetry packets or image data will be different depending on the configuration.

#### 5.8.2.7 Region of Interest EDR (ESR)

The Region of Interest (ROI) compression returns a region around each centroid in uncompressed format. ROIs are stored as a series of bitmapped images around the detected Centroids. The ROI of the centroids in the image are coded sequentially in the image file. The image data for one centroid is:

Bytes 0-1: The X coordinate of the upper left corner of the centroid.

Bytes 2-3: The Y coordinate of the upper left corner of the centroid.

Bytes 4-5: The width of the ROI (ROI\_W)

Bytes 6-7: The Height of the ROI (ROI\_H)

Bytes 8-N: The image data of the ROI starting from upper left corner.  $N = \text{ROI\_W} * \text{ROI\_H}$

### 5.8.3 SHERLOC

All Sherlock Engineering EDR products are pass-through products that have been renamed according to EDR naming convention and generated detached ODL label. The ODL label describes the DPO list in addition to the other label items.

Note that these are operational products only, and are not archived.

#### 5.8.3.1 Debug Info (ED1)

Sherloc debug info EDR is simply a pass-through of raw telemetry in binary format.

#### 5.8.3.2 Directory Dump (ED2)

Sherloc directory dump EDR is simply a pass-through of raw telemetry in binary format.

#### 5.8.3.3 Image Status (ED3)

Sherloc image status EDR is simply a pass-through of raw telemetry in binary format.

This document has been reviewed and determined not to contain export controlled technical data.



#### **5.8.3.4 Zstack List (ED4)**

Sherloc z-stack list EDR is simply a pass-through of raw telemetry in binary format.

#### **5.8.3.5 Memory Dump (ED6)**

Sherloc memory dump EDR is simply a pass-through of raw telemetry in binary format.

#### **5.8.3.6 Built in Self-Test (ED7)**

Sherloc BIST EDR is simply a pass-through of raw telemetry in binary format.

#### **5.8.3.7 Util Test (EUT)**

Sherloc util-test EDR is simply a pass-through of raw telemetry in binary format.

### **5.8.4 NavMap EDRs**

The NavMap products are engineering products used to assess the performance of the rover's on-board navigation software. All 3-char product identifiers starting with “N” (form: Nxx) are NavMap products.

NavMaps are not currently being created by IDS, but they may be in the future. The list of NavMap products and their short descriptions is thus TBD. Most, but not all, will be images of some sort in the standard VICAR/ODL format with varying data types, sizes, and numbers of bands. Because these are detailed engineering products, no further description of them is provided in this document. They will not be archived with PDS.

## **5.9 Other EDRs**

### **5.9.1 Movie EDRs**

The EDL cameras can produce movie files in MPEG format. For PDS, the MPEG file is delivered as a supplemental product, with the “science” data being extracted individual frames. For the first PDS delivery, every 10<sup>th</sup> frame is extracted. This may be changed to every frame in the future. Note that many of the frames have also been sent as lossless versions, independent of the movie files. These can be distinguished using the Compression field in the filename.

### **5.9.2 EDL Microphone**

The EDL camera suite includes a microphone. The audio data is delivered in an uncompressed WAV file format that is compatible with PDS. Operationally the audio files are also created in .flac format, but these are not delivered to PDS. The WAV format contains regular audio samples at the specified sample rate. The EDRs are typically 1-second clips; these are reassembled at the FDR stage.

## **5.10 EDR Product Validation**

Validation of the M2020 EDRs will fall into two primary categories: automated and manual. Automated validation will be performed on every EDR product produced for the mission. Manual validation will only be performed on a subset.

Automated validation will be performed as a part of the archiving process and will be done simultaneously with the archive volume validation. Validation operations performed will include such things as a validation of the PDS syntax of the label, a check of the label values against the database and against the index tables included on the archive volume, and checks for internal consistency of the label items. The latter include such things as verifying that the product creation date is later than the earth received time, and comparing the geometry pointing information with the specified target. As problems are discovered and/or new possibilities identified for automated verification, they will be added to the validation procedure.

Manual validation of the images will be performed both as spot-checking of data through-out the life of the mission, and comprehensive validation of a sub-set of the data (for example, a couple of days' worth of data). These products will be viewed by a human being. Validation in this case will include inspection of the image or other data object for errors (like missing lines) not specified in the label parameters, verification that the target shown / apparent geometry matches that specified in the labels, verification that the product is viewable using the specified software tools, and a general check for any problems that might not have been anticipated in the automated validation procedure.

There is also an implicit validation in that most of the products generated were actually used during operations, for a variety of purposes including operating the rover and doing science. Invalid EDRs that show up during use are dealt with in operations via means such as retransmitting a bad product or reprocessing data.

## 6. FDR PRODUCT SPECIFICATION

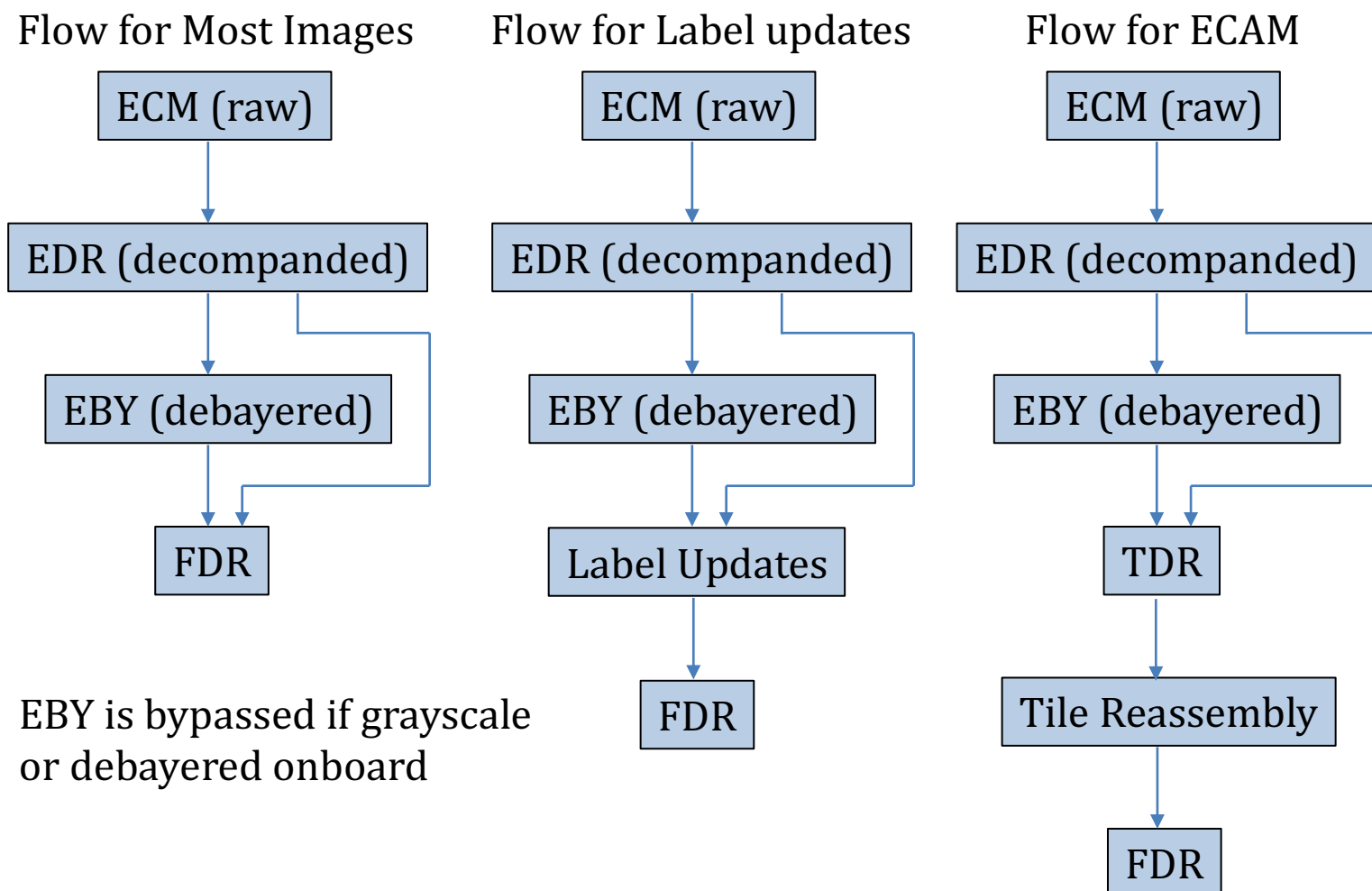
Fundamental Data Record (FDR) data products described in this document will be generated by IDS personnel using the Mars Suite of VICAR image processing software within IDS at JPL. The FDR product type is new for M2020 and represents the “base image” from which all other RDRs derive.

The primary characteristic of the FDR is that it is in a consistent representation, unlike the various EDR types. All transformations that were done to move the image from the spacecraft to the ground have been reversed (to the extent possible – some transformations are lossy), leaving either a grayscale or color image at the natural bit depth of the instrument. In particular, compression, companding, the Bayer pattern, and the effects of ECAM tiling have been removed. The result is an image that can be used as the basis for all RDR processing without regard to how it was transmitted from the spacecraft.

### 6.1 FDR General Processing

Processing for each FDR may be different for each image or instrument. In general, the FDR product is derived (copied) from either the EDR or the EBY product types. At all times the EDR type will be available. The FDR maintains a consistent starting point for all RDR processing. In the special case of ECAM, the FDR will be a reassembled image from all available fundamental tile data records (TDR) from a single image pointing. The FDR will be upscaled to the highest resolution (downsample) TDR available in the reconstruction. The ECAM FDR Processing is described in detail in section 6.2.

In a few cases, the FDR serves as a place to install metadata that was not available at the time the EDR was created, such as when the information comes down in a separate data product. Figure 6-1 diagrams the FDR creation process for each of these three cases.



**Figure 6-1: Flow Diagram for different paths to generate FDR**

## 6.2 ECAM FDR Processing (Reconstructed Images)

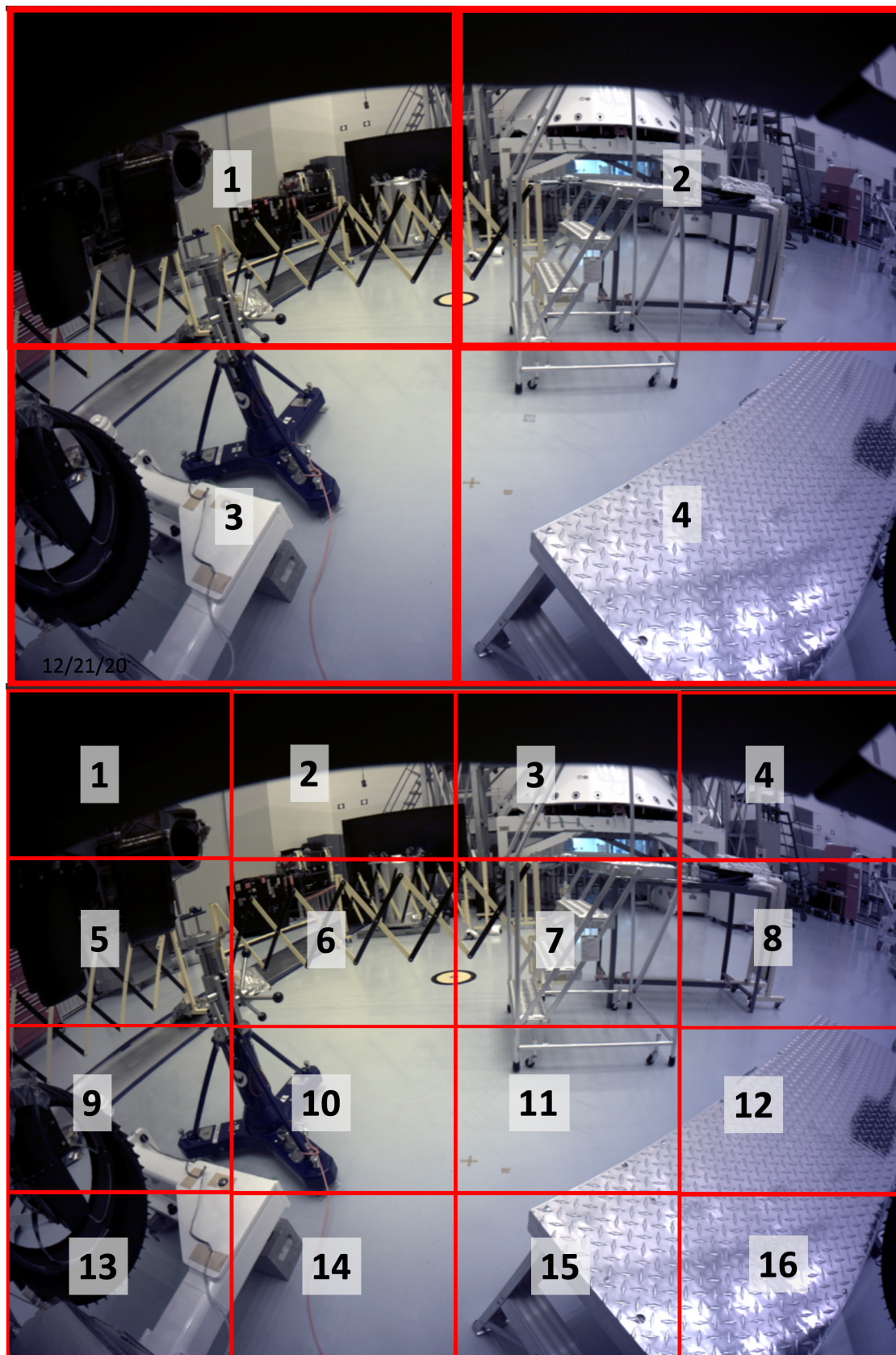
The Engineering Camera Suite on M2020 received a significant upgrade from MSL to a color (Bayer filter) 20 megapixel CMOS detector. Due to legacy (MSL) hardware restrictions, the M2020 RCE can only process ~1 megapixel at a time. Each individual portion of the detector that is readout is referred to as a “tile”, which is described in Section 5.6.1.1. The maximum tile dimensions that can be read from the camera hardware is 976 lines by 1296 samples. This means that to downlink a complete image at full-resolution, the FSW has to read out 16 separate tiles (960 lines x 1280 samples, plus an 8-pixel overlapping border on each side) and package them individually for downlink to the ground. Additionally, there are 4 flavors of readout modes from the camera hardware, all relating to the level of downsampling used to acquire the tile: 1x1, 2x2, 4x4, and 8x8. A single exposure can include any or all of the readout modes, which will have to be accounted for on the ground and reassembled (reconstructed). The IDS pipeline will collect all of the TDRs which form a single exposure and reconstruct the image based on the tiles. As more tiles become available on the ground, the reconstructed image is updated with the new information. There are several nuances to the reconstruction, all of which are described by the “camera specific” portion of the filename.

### 6.2.1 Tile Reconstruction Types

There are two possible reconstruction types: single-resolution and multi-resolution reconstructions. Single-resolution reconstruction (see Figure 6-2) contains tiles from the same exposure that are all the same downsampling resolution (1x1, 2x2, 4x4, or 8x8) and is denoted in the camera specific portion of the filename with the character “A” (see Section 19.1). Multi-resolution contains all resolutions from the same exposure, combined into the same reconstruction and is denoted in the camera specific portion of the filename with the character “M” (see Section 19.1). The multi-resolution will be upsampled to match the highest resolution tile in the reconstruction. For example: a 4x4 color full-fov image is acquired to capture a scene. Additionally a row of high-resolution (1x1) tiles is also read out to highlight an interesting outcrop. The 4 high-resolution tiles will be reconstructed with the 4x4 tile and upsampled to match the resolution of the 1x1 tiles.

#### 6.2.1.1 Reconstruction Counter

At the end of each downlink, all available tiles from a single-exposure will be combined to make the single and multi-resolution reconstructions. As more tiles are returned in subsequent passes, they will be combined into a new reconstruction, possibly changing either the color or the geometry. When a new tile is added that changes either the geometry or color of the reconstructed image, the use of a reconstruction in the camera specific portion of the filename is used to “version” the reconstruction. Differentiating between reconstruction geometries is necessary for proper stereo processing. The reconstruction counter is described in Section 19.1.



**Figure 6-2: ECAM Example of single downsampling resolution.**

**Above: 2x2, Below: 1x1.**

This document has been reviewed and determined not to contain export controlled technical data.



## 6.2.2 ECAM Fundamental Tile Data Record (TDR)

The TDR represents the “base image” from which all other tile RDRs derive. The TDR is the same concept as the FDR, but at the tile level for the Engineering Cameras and provides a consistent starting point for all tile RDR processing. There are several nuances to tiling, as described in Section 5.6.1.1, that need to be removed to enable correct reconstruction of the “full” image that makes the reconstructed FDR.

## 6.3 Label Updates

In a few cases, the FDR has updated metadata as compared to the EDRs. When EDRs are processed by the pipeline, they must be standalone. Information provided by a separate data product may not be available, because data products come down in unpredictable order (despite the use of downlink priorities, any given downlink can be missed or corrupted). The potentially missing metadata is not important enough to hold up production of the EDRs themselves – which is often the only indication the observation occurred. However, it is reasonable to hold up production of the FDR while waiting for this additional metadata - the bulk of the data is available in EDR form and the lack of a corresponding FDR serves as a flag that the metadata is missing.

### 6.3.1 PIXL Label Updates

The location of the arm (PIXL\_TOOL coordinate system) does not come down with the MCC image. It comes instead in a separate PixlTracking data product. Most of the time, PixlTracking will arrive before the MCC image, in which case the EDR products (ECM/EDR) will contain all the information. However, sometimes the PixlTracking product will be delayed. In this case, the EDR products will not have the tool coordinate system, and thus also an invalid camera model. When this happens, production of the FDR is delayed, until such point as the appropriate PixlTracking information becomes available. This usually happens by the end of the communications pass, or in the subsequent few passes. However, if the PixlTracking product is permanently lost, then the corresponding FDR may never be created.

### 6.3.2 LCAM Label Updates

The Lander Vision System camera takes images during EDL (Entry, Descent, and Landing). It is used onboard for navigation during landing. The images that are returned contain metadata about where the rover was in its descent at the time the image was taken. When the images are first downlinked, this information is produced by the onboard LVS Kalman filter, so that is what is in the EDR labels.

Following the landing, data from the DIMU (Descent Intertial Measurement Unit), and optical observations were used to reconstruct the EDL trajectory [Serricchio2021]. The descent stage trajectory reconstruction was initialized from the landing site using the inflight-navigated position and velocity, which included landing radar measurements. The final position was shifted to the landing location as determined by rover navigation camera orthophotos co-referenced to a HiRISE image derived map, and an attitude initialization correction of 33mdeg applied. The descent stage trajectory was then backward integrated using only the (time-corrected) DIMU data in the inertial frame, and transformed to various other coordinate frames of interest. In addition, data from the RIMU (Rover IMU) was used to reconstruct the rover trajectory during the skycrane phase. The rover trajectory was initialized using the reconstructed descent stage trajectory and the nominal descent stage / rover alignment determined from metrology.

The LCAM poses were derived by combining the descent stage and rover reconstructed trajectories queried at the image exposure midpoint with the LCAM-to-rover and LCAM-to-descent frame alignments determined by metrology. (Team credit: F. Serricchio, M. San Martin, P. Brugarolas, J. Casoliva, N. Trawny, J. Montgomery).

This information is what appears in the FDR labels. Specifically, it is the CINT\_COORDINATE\_SYSTEM group that is updated, along with the camera models. Note that the original estimation in the EDRs is replaced by the updated estimation in the FDR – the original is not maintained in the FDR label.

## 6.4 Audio Data

Audio data from the EDL microphone typically comes down in 1-second clips. These are marked as EDRs (type code EAU). These clips are reassembled into the original, longer clip at the FDR stage (type code FAU). Note that each clip has an index number in the camera-specific part of the filename; these are used to assemble the clips in order. If an index is missing, a different reconstructed clip is created starting after the break. Thus the index in the FAU filename indicates where the clip starts in the overall sequence.



## 7. PIXEL-BASED IMAGE PROCESSING

### 7.1 Radiometric Correction

Radiometric correction is the process of converting the image to a calibrated unit that represents the light entering the lens of the camera. It thus provides a representation of the scene as it exists, independent of the camera. It should not be confused with photometric correction, which takes into account lighting and geometry in order to describe the objects in the scene, independent of how they were observed.

While there are many different kinds of radiometric units, only three have relevance to M2020 processing.

1. Instrument Effects: These are still in units of DN, but remove certain instrument effects.
2. Spectral Radiance: Units of  $W/m^2/nm/steradian$ . This is the primary radiometric unit used.
3. Radiance Factor: Dimensionless quantity showing I/F, for Incidence/Flux.

Within each kind, the values may be represented as floating-point values (containing the actual unit value), or scaled to integers for ease of manipulation. If scaled, the scaling is shown by the `RADIANCE_SCALING_FACTOR` and `RADIANCE_OFFSET` values in the VICAR/ODL label, or the `<scaling_factor>` and `<value_offset>` attributes in the Array class in PDS4.

#### 7.1.1 MIPLRAD

MIPLRAD is the radiometric correction algorithm for any camera systematically performed by MIPL (IDS at JPL) in the IDS pipeline. The resulting rad-corrected RDRs are integrated into terrain mesh products used for traverse planning, and mosaics used for situational awareness and science analysis during operations. This method is typically less precise than the methods used by the science teams.

In summary, MIPLRAD first backs out any onboard flat field that was performed. It then applies the following corrections: bias or shutter subtraction, flat field, exposure time, temperature-compensated responsivity, bits per pixel, and zoom. The result is calibrated to physical units of  $W/m^2/nm/sr$ . The actual algorithm and equations used for MIPLRAD are shown below. Not every step applies to each camera type. Each correction is applied in sequence, to every pixel. In each step, **img** is the input while **img'** is the output (which becomes the input **img** of the next step).

All coefficients are in the calibration files, described in Appendix C.

Some steps are dependent on temperature, focus, or zoom. The focus value comes from `INSTURMENT_FOCUS_POSITION` or, if not available, `FOCUS_POSITION_COUNT`. The zoom value comes from `INSTRUMENT_ZOOM_POSITION` or, if not available, `ZOOM_POSITION_COUNT`. The temperature is determined from the `INSTRUMENT_TEMPERATURE` label using the following guidelines, where the first valid temperature found (0.0 is ignored as no-reading, and >50C is considered broken) is the one used:

- a) Use the CCD, mounting plate, or optics box temp of said camera
- b) Use the CCD, mounting plate, or optics box temp of left/right partner

This document has been reviewed and determined not to contain export controlled technical data.

- c) Use the CCD, mounting plate, or optics box temp of similar camera (e.g. nav vs haz)
- d) Use the electronics temperatures, in similar order

The ZCAM and Watson cameras have variable flat fields based on temperature, zoom, or focus. These “\*.varflat” files in the calibration directory specify the variable being used, and the value that each separate flat field uses. It uses the closest flat field to the given variable value – flats are not interpolated between values.

### 7.1.1.1 Basic MIPLRAD

The basic MIPLRAD algorithm is used by the Engineering Cameras, and all other cameras not listed below.

1. For the Engineering Cameras only, if on-board flat-fielding has been applied, it is backed out according to the parameters in FLAT\_FIELD\_CORRECTION\_PARM in the image label.

$$\begin{aligned}
 r &= (x - a)^2 + (y - b)^2 \\
 ff_{x,y} &= 1 + cr + dr^2 + er^3 \\
 img'_{x,y} &= img_{x,y} / ff_{x,y}
 \end{aligned}$$

where  $a-e$  are the coefficients of FLAT\_FIELD\_CORRECTION\_PARM.

2. A bias value is subtracted from the image, which takes the place of shutter subtraction:

$$img'_{x,y} = img_{x,y} - bias$$

3. The flat field is then applied. This removes vignetting and other lens patterns, as well as partially compensating for bad pixels. There are separate flat fields for Bayer-encoded and decoded images, but in general radiometric correction must be performed on decoded images. For Watson, the varflat mechanism is used to choose the closest flat field based on focus. The x and y coordinates are adjusted based on downsampling and subframing to find the corresponding pixel in the flat field, then the DN is divided by the flat field value (obtained by pre-launch calibration or analysis of sky images):

$$img'_{x,y} = \frac{img_{x,y}}{ff_{x',y'}}$$

4. Exposure time is then removed. Exposure time comes from the EXPOSURE\_DURATION label, converted to seconds. An optional exp\_overhead value can be used to account for overhead that is not included in exposure time (but is generally not used):

$$img'_{x,y} = img_{x,y} / (exposure + exp\_overhead)$$

5. The responsivity is computed next. Responsivity is the primary calibration component, converting DN to Spectral Radiance. Units are W/m<sup>2</sup>/nm/sr/(DN/sec). Responsivity can be temperature dependent. The temperature is combined with responsivity parameters  $r_0, r_1, r_2$ , which were derived from ground calibration according to the formula below. For color images, the responsivity is computed separately per color band.

$$r = r_0 + r_1 t + r_2 t^2$$

6. For SuperCam RMI, the responsivity is then modified based on the number of bits. The calibration coefficients are set assuming 12-bit data. If the data is only 10 bits, the responsivity must be scaled by a factor of 4. This is done by comparing the number of bits in the image with the nominal number of bits from the calibration file.

$$r' = r * 2^{(12-\text{bits})}$$

7. The responsivity is then applied to the image. Note that it is this final  $r''$  value that ends up in the image label.

$$\text{img}'_{x,y} = \text{img}_{x,y} * r''$$

8. Finally, the result is (optionally) converted to integers using the RADIANCE\_OFFSET and RADIANCE\_SCALING\_FACTOR keywords:

$$\text{img}'_{x,y} = \frac{\text{img}_{x,y} - \text{offset}}{\text{scale}} + 0.5$$

### 7.1.1.2 SkyCam MIPLRAD

SkyCam uses a more complex shutter subtraction system. First, several constants are computed based on parameters in the rad cal file.  $T_c$  is the temperature of the CCD,  $T_e$  is the temperature of the electronics, and  $V_{\text{off}}$  is from the DC\_OFFSET label:

$$B = B_0 + B_1 T_e + B_2 T_e^2 + (4095 - V_{\text{off}})$$

$$S = S_0 e^{S_1 T_c}$$

$$A = A_0 e^{A_1 T_c}$$

There are four calibration images: FF = traditional flat field, AD = Active Dark, FT = Frame Transfer, BC = Bias Column. Steps 2-4 and 7 above are then replaced by one of two formulae.

If onboard shutter subtraction was performed:

$$\text{img}'(x,y) = \frac{(\text{img}(x,y) - A * \text{AD}(x,y) * \text{exposure}) * r}{\text{FF}(x,y) * \text{exposure}}$$

If onboard subtraction was not performed:

$$\text{img}'(x,y) = \frac{(\text{img}(x,y) - B - \text{BC}(x) - S * \text{FT}(x,y) - A * \text{AD}(x,y) * \text{exposure}) * r}{\text{FF}(x,y) * \text{exposure}}$$

### 7.1.1.3 Mastcam-Z MIPLRAD

Mastcam-Z is considerably more complicated, because it takes into account zoom as well as temperature. Steps 2-7 are replaced by the below. The below describes the mechanics of the algorithm; see [Hayes2020] for an explanation of it.

For Mastcam-Z, responsivity is computed by the formula below, which also uses the reference temperature  $t_0$ , defined in the calibration file. Additionally, the responsivity parameters differ per filter.

$$r = \frac{r_0}{1 + r_1(t - t_0)}$$

The responsivity is then scaled based on the zoom factor. This scaling is based on a ratio of f/numbers between the image and the reference zoom at which the calibration data was taken (100mm zoom).

$$r' = \frac{r}{(f/f_0)^2}$$

The relationship between zoom and f/number is defined in the table named by the zoom\_fnum\_ref parameter in the rad cal file. This table is a simple CSV file with zoom motor count and f/number (and an optional 3<sup>rd</sup> column which is ignored). The parameter  $f$  above is found in this table, while  $f_0$  is defined in the rad cal file under zoom\_fnum\_ref.

There are four calibration images for Mastcam-Z: RAD, FZOOM, STATIC, and DYN. DYN is the dynamic bias, which is a single image per filter. STATIC is the static bias, which uses the “varflat” mechanism as a function of temperature. RAD is the flat field radiometric set; it has a value for each filter but at only 3 canonical zoom positions (using the varflat mechanism). FZOOM is the continuous zoom set; it has a value for 67 different zoom values but only at filter 0.

The flat fields are combined by taking the RAD flat for the given filter using the closest of the 3 canonical zoom positions, and then normalizing it by the ratio of actual zoom to canonical zoom using the FZOOM for filter 0. In the equation below, (x,y) were omitted for clarity; all terms are images.

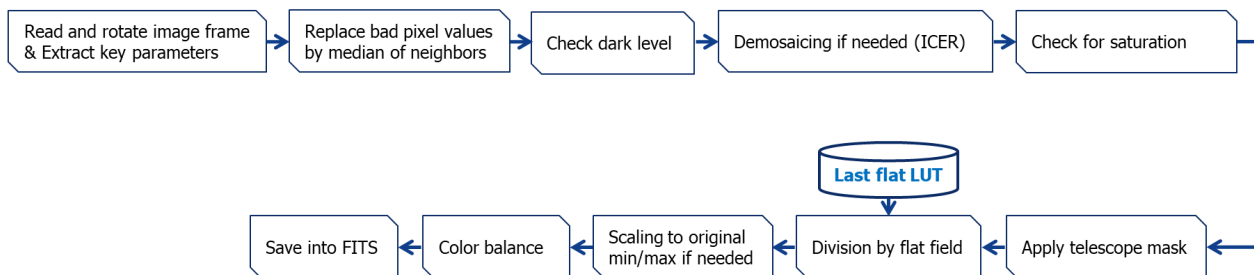
$$ff = RAD_{f,can} * \frac{FZOOM_{0,act}}{FZOOM_{0,can}}$$

The radiometric correction is then

$$img'_{x,y} = \frac{(img_{x,y} + V_{off} - STATIC_{t,x,y}) * r'}{ff_{x,y} * (exposure + DYN_{f,x,y})}$$

Where  $V_{off}$  is from the DC\_OFFSET label and  $r'$  was computed above.

## 7.1.2 CHEMRAD



**Figure 7-1: RMI Image Pipeline (simplified)**

This refers to partial radiometric correction of SuperCam RMI image data performed using an image processing toolkit developed by SuperCam Science Team members at IRAP in France. It is deemed “partial” in the sense that the raw image data are not calibrated to physical radiance units. Although the calibration portion of radiometric correction is not performed for the input EDR, other corrections are applied.

The main procedure can apply successive corrections to RMI EDR image defects by incorporating, in a specific order, a number of correction steps. A simplified processing flow is shown in Figure 7-1.

The resulting RDR product is only partially radiometrically corrected and is not calibrated into physical radiance units. Note that the default flat field LUT is derived from a lamp observation during ground testing and will be replaced by an observation of the sky in flight.

Effects not accounted for in the correction process include:

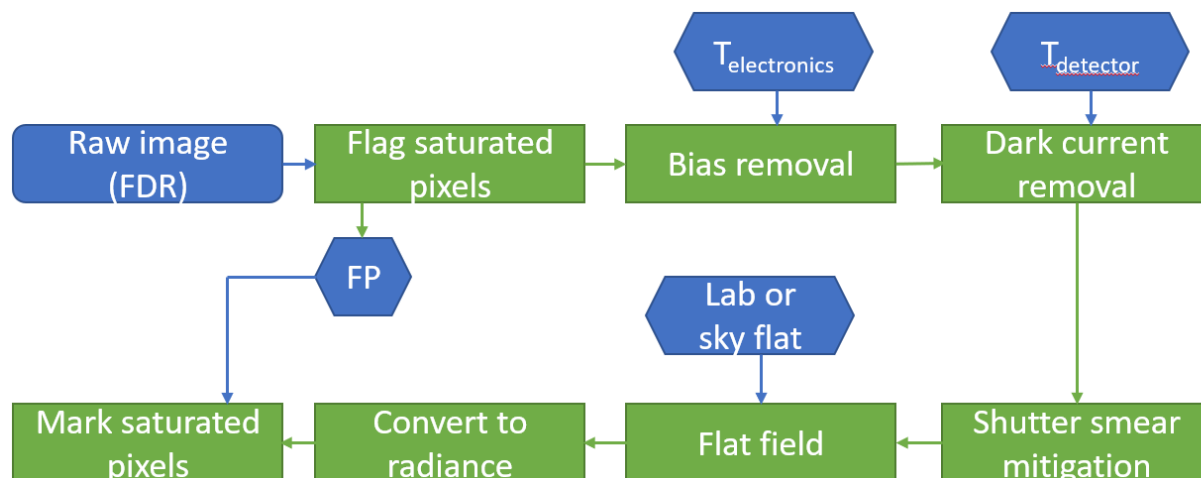
- a) Non-linearity - Each pixel of the CMOS is a potential well, in which electrons are trapped. The number of trapped electrons is an affine function of the lighting, but above a certain threshold the linearity is broken (saturation)..
- b) Direct Light - Some light from the far field can reach the detector without passing by the telescope mirrors, despite the baffle on the secondary mirror. The diffuse light is taken into account in the determination of the reference flat field, and thus rather well corrected, but the focused distant scene cannot be corrected. However, this rare and faint.

## 7.1.3 MEDA SkyCam Radiometric Processing

Initial radiometric correction of Skycam images is performed using correction factors provided by the MEDA team. Further radiometric correction is done using results from in-flight calibrations by the Skycam team at Space Science Institute (SSI) and Centro de Astrobiologia (CAB). The SSI pipeline is described below and in Figure 7-2.

Radiance-calibrated RDRs (“RAD” files) are generated from raw, 12-bit FDRs. Depending on the image acquisition mode, the image may have 1056 columns (1024 image columns with 17 bias columns prepended and 14 bias columns and an image ID column appended) or 1024 columns (image only). Saturated pixels are recorded. “Standalone” images require bias removal, frame transfer dark current removal, shutter smear mitigation, and dark current removal prior to further processing. “Brightdark” images have had a 0-sec exposure image subtracted on board and require

only the active dark current removal. If needed, bias is calculated from electronics temperature (image-only) far from the prepended columns (if available) and removed from all columns of the image; if necessary, the image is trimmed to 1024x1024. The detector temperature is used to calculate the frame-transfer dark correction if needed; temperature and exposure time are used to calculate the active dark current correction. Based on the scene content after dark removal, a frame transfer (electronic shutter) smear image is estimated, and that is subtracted from the image. At this point, the image is divided by a flat-field image and by exposure time to yield an image in DN/s. The instrument responsivity is multiplied in to convert to physical units. Finally, saturated pixels are flagged for the final product.



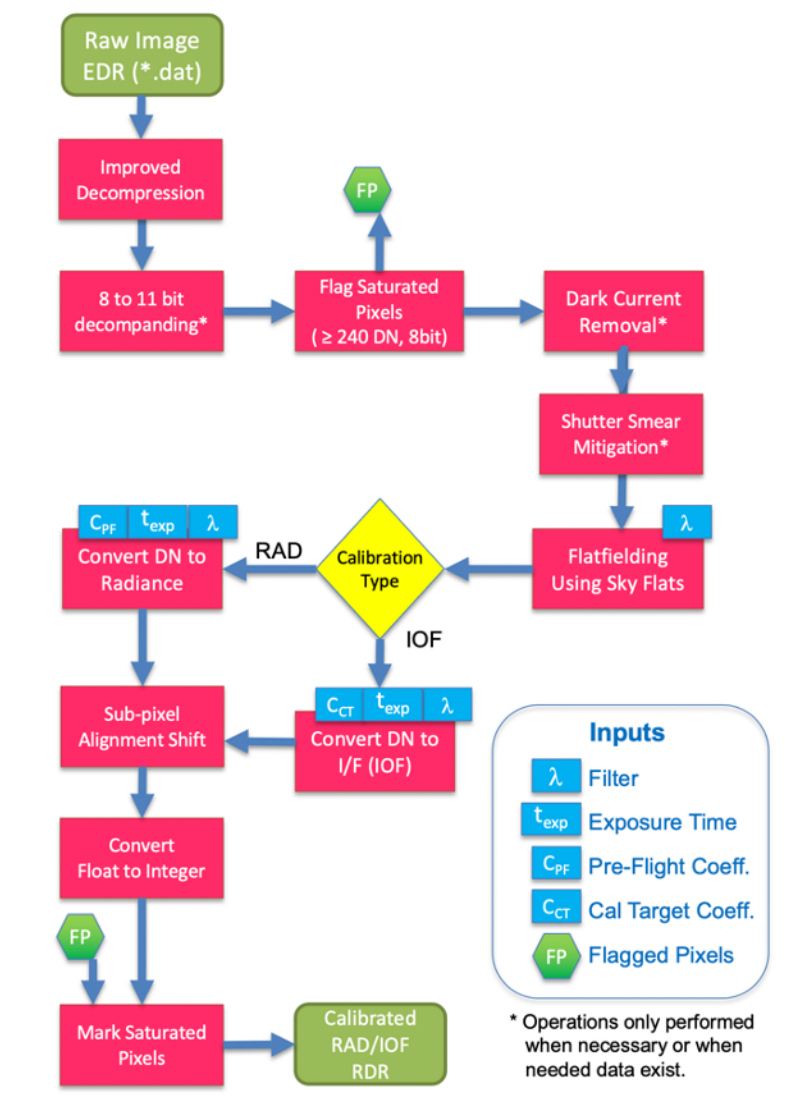
**Figure 7-2 High level flow chart describing radiance calibration of Skycam images.**

### 7.1.4 ZCAM Radiometric Processing

Radiometric correction of Mastcam-Z images is performed by the Mastcam-Z downlink team at Arizona State University (ASU). There are two general types of Mastcam-Z RDR products that are generated by the team, following the high-level flow chart process shown in Figure 7-3.

- Radiance-calibrated RDRs ("RAD" files): These RDRs are generated from raw, not geometrically linearized, EDRs. They have all the major instrumental/environmental calibrations applied, such as bias removal, dark current removal, electronic shutter smear effect removal, flat field correction, and bad pixel repair. After those corrections have been applied, the data are scaled to absolute radiance units using either pre-flight filter-dependent, zoom-dependent, and temperature-dependant radiometric calibration coefficients. The units on these files are (W/m<sup>2</sup>/nm/sr).
- Radiance factor-calibrated RDRs ("IOF" files): These RDRs are also generated from raw EDRs. They have had all the major instrumental/environmental calibrations applied, have been scaled to absolute radiance units as described above, and then have been further scaled by a set of filter-dependent reflectance calibration coefficients derived from in-flight observations of the Mastcam-Z calibration targets acquired closely in time. The values in these files correspond to radiance factor, or "I over F", where I is the radiance from the Martian scene and p \* F is the radiance from the Sun at the top of the Martian atmosphere. Since the solar radiance in the same units as the Mars scene radiance was divided out, these files are unitless, but have values in the range of 0.0 to 1.0. For example, average bright Mars soils exhibit I/F ~ 0.35 at 750 nm and I/F ~ 0.05 at 410 nm.

Additional details on the radiometric processing and calibration of Mastcam-Z images can be found in the published open-access research papers: [Bell2020], [Hayes2020], and [Kinch2020].



**Figure 7-3: High-level flow chart describing radiance ("RAD") and radiance factor ("IOF") calibration of raw Mastcam-Z EDRs, conducted at Arizona State University based on Mastcam-Z instrument and calibration target performance and characteristics described in Bell et al. (2020), Hayes et al. (2020), and Kinch et al. (2020).**

### 7.1.5 Radiometric Float to Integer Scaling

The various kinds of radiometrically corrected products are all based on physical units. They are therefore naturally floating-point values. However, floating-point images are more difficult to deal with and many image processing or display systems work only with integer images. For this reason, the floating point radiometric values are often scaled to integers (the original floating-point images are also sometimes preserved).

There are three types of scaling: static, dynamic, and per-image. All use a scaling factor and offset (which are stored in the label) to convert between integer and float:

$$\text{float} = \text{int} \times \text{scale} + \text{offset}$$

The difference is in how the scale is determined. Note that offset is always 0 for MIPLRAD but may not be for the other methods. MIPLRAD makes use of static and dynamic, while ZCAM uses per-image. Note that the CHEMRAD method results in integer values closely related to the original DN; it does not generate floating-point units so the scaling described in this section does not apply.

Static scaling uses single, predefined scaling factors, which have been selected to best represent the usual range of intensities seen across all images on Mars. Static scaling has the very useful property of being constant from image to image, thus making mosaics or meshes much easier to create. For this reason, static scaling is used for the RAD/RAS products created by IDS that are the basis of IDS mosaics and meshes. However, in exceptionally bright (specular reflection) or low-light cases, static scaling can saturate or be too dark to see.

Dynamic scaling determines the scale factors independently for each image, using a formula. This optimizes the dynamic range for each image, but makes it unsuitable for direct use in mosaics (without reconstituting the float value first) because the scaling varies from image to image. Dynamic scaling is thus better for single images, with static scaling better for mosaics. Dynamic scaling is used in the IDS RAY product, among others. The dynamic scale factor incorporates the exposure time, maximum responsivity across bands, and average flat field (called “flat\_darkening\_factor” in the calibration files). The result is that the integer image is very similar to the ILT from before radiometric correction in dynamic range, with corrections only for *average* flat field, and the ratio of responsivities across bands. However, the floating-point value can still be reconstructed using the scale factor. Note that the same scaling is used for all bands. The scale factor is computed as:

$$\text{dynamic\_scale} = \text{exposure\_time} \times \text{flat\_darkening\_factor} / \text{max\_responsivity}$$

Per-image scaling looks at the minimum and maximum values and assigns the scale and offset so as to maximize the use of the available dynamic range. This is often done on a per-band basis, so looking at images with per-image scaling without compensating for the scale/offset factors may result in a perceived color shift of the image. This is the method used by the ZCAM radiometric processing.

### 7.1.6 Zenith Scaling

Radiometric correction results in a product that describes how much light entered the camera. However, lighting varies due to solar elevation and atmospheric opacity, among other things. These photometric effects mean that images taken at different times of day have different intrinsic intensities, making them difficult to combine into a mosaic or mesh, even after radiometric correction. Zenith scaling is a limited form of photometric correction that helps even out the brightness for meshes and mosaics.

Zenith scaling attempts to reconstruct what the radiometric product would look like if the sun were at the zenith on a clear day. It takes into account the solar elevation and “tau”, which is a measure of atmospheric opacity. The formula is:

$$\mu = \sin(\phi)$$

$$f = \mu e^{\left(\frac{-(\tau - \tau_{ref})}{6\mu}\right)}$$



where  $f$  is the correction factor,  $\phi$  is the solar elevation,  $\tau$  is the solar opacity (tau), and  $\tau_{ref}$  is the reference tau, what is defined to be a “clear day”. For M2020 the reference tau is set to 0.3. Note that if the solar elevation is less than a minimum (a tunable parameter which defaults to 5 degrees), it is set to that minimum in the formula above.

The radiometric product is divided by this correction factor in order to create the zenith-scaled product. This has the effect of boosting the brightness of images taken earlier or later in the day.

## 7.2 Color Correction

For the cameras with Bayer-pattern color filter array (CFA), in order to display color images correctly on a computer monitor, raw color images are converted from the instrument color space (iRGB) to other color spaces by multiplying the R, G, and B bands of the iRGB image by a 3x3 color conversion matrix. The color transform step is performed on the ground as part of the automated MIPL pipeline processing.

During the M2020 operational mission, raw iRGB images will be converted to two main color spaces: the device-independent CIE (Commission internationale de l'Eclairage) XYZ color space [CIE1932] and the IEC sRGB color space [ITU2015], for use on display devices. A third colorspace, CIE xyY, is related to CIE XYZ, but it separates the luminance component (Y) from the color, leaving a 2-dimensional chromaticity component (xy). For more information on the CIE spaces the reader is referred to [Wyszecki1982]. A third class of colorspace are white-balanced color spaces. Images are white balanced by adjusting the primary (R, G, B) colors such that white objects in the scene are represented as white in the image. This is typically done by applying a 3x1 matrix multiplication (or in practice, a 3x3 matrix with non-zero coefficients along the diagonal, and zeroes elsewhere). During the surface mission white-balanced iRGB and sRGB images were generated for the operations team.

The RGB images discussed above are created and converted in linear brightness space. However, most display systems (including the sRGB standard) expect images to be gamma-encoded. Therefore, the sRGB and white-balanced images are gamma-encoded for display (which also converts them from 12 bits per pixel to 8 bits per pixel) using the function described in [Maki2018] [Abarca2019]. It is these gamma-encoded products that should be used on most display systems. The CIE XYZ and CIE xyY products are both in floating-point format, so no gamma encoding is done on those.

Independently of all the above, there are three types of scaling that are performed on color products, which is indicated by the first letter of the RDR type code.

Those that start with C are exposure-time compensated (or normalized) color products. These are derived directly from the radiometrically corrected product (RAF), with a static scaling factor to bring the DN values of the color products into the 0-4095 range. This means the C\*\* products are similar to the “static scaling” rad products defined in Section 7.1.5, in that the color is consistent from image to image in a mosaic, but it’s possible the dynamic range of an individual image may be too bright (saturated in places) or dark to see well.

The three-letter product codes for color products follow a standard pattern:

Character 1: Color Indicator

- C = Exposure Time Compensated (or Normalized) Color
- D = Raw DN Color
- Z = Zenith-Scaled Color
- M = Masked Color

Character 2: sRGB or White Balance Indicator

- N = Neither (N/A, or XYZ/xyY color spaces)
- S = sRGB, as measured by the camera
- P = White-balanced sRGB, Mars illuminant
- W = White-balanced iRGB, Mars illuminant

Character 3: Data Type

- D = 15-bit integer, linear space
- S = 12-bit integer, linear space
- F = Float, linear space
- B = Byte, linear space
- G = Byte, gamma correction applied

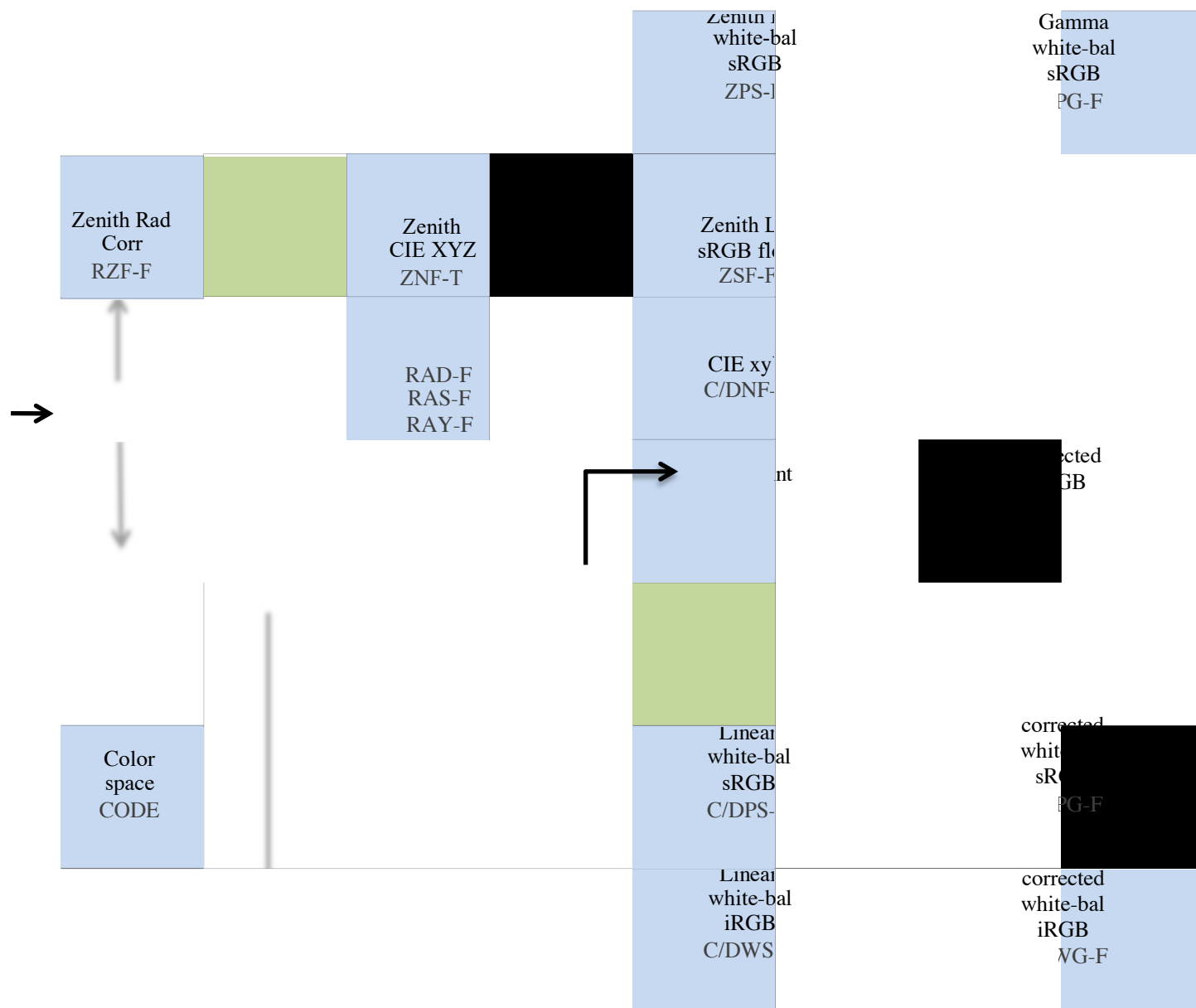
Those that start with D are raw DN color products. These are created in the same way as the C\*\* products, except that the scaling factor is the exposure time. This has the effect of removing exposure time compensation (it was included in the RAF source product), so the DN values are based on the raw DN's that came from the spacecraft (modified by the average flat field). In this way they are similar to the “dynamic scaling” rad products from Section 7.1.5. The dynamic range of each image is optimized, but the overall intensity will vary from image to image in a mosaic.

Those that start with Z are zenith-scaled color products. They are just like C\*\* except they are derived from the zenith-scaled radiometry (RZF, Section 7.1.6) rather than RAF. They are thus the best products to use for mosaics or meshes (better even than C\*\*) because they are compensated (in an approximate way) for lighting conditions.

Finally, those that start with M are masked color products. These are one of the other types that has had a mask applied to it to remove the arm from IDC images. The M\*\* type can be based on any of the other (C/D/Z) color types; see the label for which it is.

Only a few of the possible product types are generated systematically by the pipeline (marked by green or yellow in Table 17-2). Any type may be created as a special product, however.

Figure 7-4 shows the pipeline processing flow for radiometric and color products. Note that floating-point versions of the products are used at all intermediate steps (except for the linear to gamma conversion) to avoid clipping due to saturation. Additional clipping in the linear to gamma conversion is not possible given how gamma conversion is defined. The same flow is used for C\*\* and D\*\* products; the only difference is the parameter given to marscolor when creating the CNF/DNF products.



**Figure 7-4: Color and Radiometric Processing Flow. The code contains the 3-letter product type and the Filter/Color code (indicating the color type)**

The RGB images discussed above are created and converted in linear brightness space. However, most display systems (including the sRGB standard) expect images to be gamma-encoded. Therefore, the sRGB and white-balanced images are gamma-encoded for display (which also converts them from 12 bits per pixel to 8 bits per pixel) using the function described in [Maki2018],[Abarca2019]. It is these gamma-encoded products that should be used on most display systems. The CIE XYZ and CIE xyY products are both in floating-point format, so no gamma encoding is done on those.

## 7.4 Photometry

The IEP type contains incidence, emission, and phase angles for each pixel for use in photometry work. It is a 3-band float product derived from UVW in the order (incidence, emission, phase). It can also be filled (IEA). This product is not made by the IDS pipeline but is occasionally made for photometric analysis.

## 8. GEOMETRIC AND STEREO PROCESSING

### 8.1 Stereo Pair Matching

In nominal operation, all of the stereo cameras – Navcams, Hazcams, and Mastcam-Z – acquire stereo images simultaneously. This means the left and right eyes are taken at the same instant and share the same SCLK as reflected in the filename. This allows stereo pairs to be easily matched, by simply looking at the filename for matching SCLK and separate eyes. In the case of the Mastcam-Z, the images may be acquired some milliseconds apart, but the FSW assigns the same Data Product Validity Time (DVT) to both eyes, so that is what is used in the filename. Other SCLKs in the label reflect the actual exposure time.

However, there are situations where stereo images are not acquired simultaneously. This is particularly common with the Mastcam-Z (Section 14.1) but there are other use cases, such as long-baseline stereo (where the rover moves between frames), arm camera stereo (where the arm moves a non-stereo camera to achieve a stereo baseline), and even cross-instrument stereo.

The Stereo Counter flag in the filename (Table 18-1) indicates the stereo status. Standard simultaneously-acquired stereo images have an underscore “\_” in the Stereo Counter field. Repointed stereo has a character other than “\_”.

For repointed stereo, the Stereo Counter field is a *counter* – it indicates that there is an off-nominal partner, and groups all the RDRs related to that partner (they all share the same counter value) but it does not *in any way* indicate what the partner is. Specifically, you cannot look at image A with a counter of “2” and image B with a counter of “2” and assume they match – they will not, in general. The combinatorial explosion problem simply makes it impractical to use the Stereo Counter in this way (we would run out of available values on a reasonably simple panorama). So the Stereo Counter merely groups all RDRs of one image that use one specific partner, without identifying the partner.

In order to determine what the stereo partner is, one must look at the label. In VICAR/ODL, the STEREO\_PRODUCT\_ID provides the product id (filename minus extension) of the partner. In the PDS4 labels, the <img\_surface:Stereo\_Product\_Parameters> class provides a PDS4 pointer to the stereo partner.

### 8.2 Camera Models

Camera models describe the geometry of a camera. They are used to convert from XYZ coordinates in the world to line, sample coordinates in an image, or to convert from line, sample to a view ray (origin and direction) in 3D space (a single model cannot determine range). Camera models are critical for any kind of geometric processing. See [Yakimovsky1977], [Gennery1986], [Gennery2002] and other resources listed in the Reference Section.

Camera models consist of a set of equations, and numeric parameters used in those equations. As used in this document “Camera model” generally means the set of numeric parameters, which appears in the image label.

This document has been reviewed and determined not to contain export controlled technical data.

There are three families of camera models used in M2020, described below. The CAHV family is by far the most important, being used in operations in every image. The other two have niche uses.

Camera models have both “intrinsic” parameters, which are inherent to the camera itself, such as focal length and pixel scale, and “extrinsic” parameters, which describe how the cameras relate to the world (position and orientation). In the CAHV family, these are mixed in the parameters; they are not cleanly separated.

The calibration camera model is created at a known pointing. This is then transformed to point the camera in the desired direction (i.e. the extrinsics are modified), which creates a model specific to the given image. This is what is stored in the label. See Section 8.3.

An important point regarding camera models and subpixels: the integer is defined to be in the *center* of the pixel. Thus a pixel actually covers a space of  $x-0.5$  to  $x+0.5$ . The math for all three model types describe below uses 0-based pixel numbering. Therefore when using the models directly, the leftmost/topmost pixel has a central coordinate of (0.0, 0.0) meaning that the upper left corner of the pixel is at (-0.5, -0.5). However, per PDS convention, in the label, all offsets are 1-based, so the upper left corner of the upper left pixel is actually (0.5, 0.5) and the bottom right corner of that pixel is (1.5, 1.5). It is important to recognize this difference between 0-based and 1-based numbers and use them appropriately.

## 8.2.1 CAHV family of camera models

The CAHV family of camera models is the primary type of camera model used in Mars 2020. It is a common model used in robotics at JPL. The model name is based on the one-letter terms used to name the parameters for each model. There are three models in the CAHV family. See the references for a complete treatment as well as the mathematics behind each model, but the parameters are briefly described below.

### 8.2.1.1 CAHV

The CAHV model [Yakimovsky1977] describes a linear, perspective-projection camera, otherwise known as a pinhole camera. Given a 3D point  $P$ , the image coordinates may be computed as:

$$sample = \frac{(P - C) \cdot H}{(P - C) \cdot A}$$

$$line = \frac{(P - C) \cdot V}{(P - C) \cdot A}$$

**C: Center Vector** – The 3D position of the center of the entrance pupil (technically, the image of the entrance pupil, where it would appear to an external observer looking into the lens). Often thought of as the “position” of the camera.

**A: Axis Vector** – A unit vector normal to the image plane, pointing outward. This is not what most people mean by the pointing direction of the camera, although it sometimes serves as a proxy. The pointing direction is more properly the ray pointing from the center of the image through the C point. Note that for the SuperCam RMI model in particular, the A vector is significantly different from the pointing direction.

**H: Horizontal Vector** – A vector pointing roughly rightward in the image, but actually a composite of several characteristics. A projection of this vector into the image plane, called  $H'$ , is thought of as showing how the rows are oriented in 3D, but in truth is a vector orthogonal to the columns. The magnitude of this projection is the Horizontal Scale,  $H_s$  (this is not the focal length). A projection of  $H$  parallel to  $A$  has a magnitude called the Horizontal Center,  $H_c$ . This is the horizontal image coordinate directly under the lens center ( $C$ ) when moving parallel to  $A$ . This could be outside the bounds of the image for some models (such as RMI). These quantities are related via:

$$\begin{aligned} H' &= H - (A \cdot H)A \\ H_s &= |A \times H| \\ H_c &= A \cdot H \end{aligned}$$

**V: Vertical Vector** – Entirely analogous to  $H$  except in the vertical direction, pointing roughly downward.

### 8.2.1.2 CAHVOR

The CAHVOR model [Gennery1991] is an extension of CAHV that adds radial (barrel or pincushion) distortion on top of the linear model. This better models lenses with wider fields of view.

**O: Optical Vector** – A unit vector, pointing outward, that is the symmetry axis for the radial distortion.  $O$  is generally somewhat close to  $A$ , but is rarely the same.

**R: Radial Distortion Terms** – The coefficients of a polynomial function used to model radial distortion. See the references.

### 8.2.1.3 CAHVORE

The CAHVORE model [Gennery2002] is a more general model that can be used to model fisheye lenses. It implements a moving entrance pupil. According to the parameter list it should properly be called CAHVORETP but this is never done in practice.

**E: Entrance-pupil terms** – The coefficients of a polynomial function used to model movement of the entrance pupil. See the references.

**T: Type** – Defines the type of CAVHORE model. 1=Perspective-projection, which is similar to CAHVOR. 2=Fish-eye lenses. 3=General, a generalized model that incorporates aspects of both 1 and 2.

**P: Parameter** – Scalar parameter characterizing type 3. If the parameter is 1, the model is the same as type=1. If the parameter is 0, it is identical to type=2. Values in between incorporate aspects of both.

## 8.2.2 PSPH Camera Model

The Planospheric (PSPH) camera model [Litwin2020] provides better fisheye-image rectification prior to 1D stereo correlation, using a unit projection sphere rather than an image plane, whose center coincides with the projection point ( $C$ ) of the virtual lens.

The PSPH model is used in FSW for onboard correlation of the Hazcams. It is a virtual model, in that no real camera is calibrated from this model; it is created transiently as needed. It is not as of this writing used in the ground system, but it could appear in labels at some point. The model parameters are briefly described below.

Image coordinates:

$x$  = column, horizontal coordinate  
 $y$  = row, vertical coordinate

Model parameters:

$\mathbf{c}$  = 3D position of the sphere center  
 $\mathbf{a}_x$  = unit column-plane rotation axis, passing through the sphere center, typically vertical and pointing down so that positive rotations (by the right-hand rule) will rotate the forward half of the plane in the (rightward) direction of increasing column (as projected on the forward hemisphere)  
 $\mathbf{a}_y$  = unit row-plane rotation axis, passing through the sphere center, typically horizontal and pointing left so that positive rotations (by the right-hand rule) will rotate the forward half of the plane in the (downward) direction of increasing row (as projected on the forward hemisphere)  
 $\mathbf{n}_x$  = unit normal vector to the column plane when  $x$  equals zero, pointing in the same direction as the cross product of  $\mathbf{a}_x$  with an outward-pointing vector that also lies in the plane  
 $\mathbf{n}_y$  = unit normal vector to the row plane when  $y$  equals zero, pointing in the same direction as the cross product of  $\mathbf{a}_y$  with an outward-pointing vector that also lies in the plane  
 $s_x$  = column scale factor to convert between  $x$  coordinate and rotation around  $\mathbf{a}_x$ , expressed in radians/pixel  
 $s_y$  = row scale factor to convert between  $y$  coordinate and rotation around  $\mathbf{a}_y$ , expressed in radians/pixel

### 8.2.3 Photogrammetric Camera Model

The photogrammetric camera model [Kraus2007] is commonly used in the photogrammetric community and is in particular implemented by [OpenCV]. It is used by the Mastcam-Z science team for some of their processing. As of this writing the labels have not been fully defined but in general the parameters are:

The intrinsic parameters of the camera:

$f_x, f_y$  – Focal lengths expressed in pixels  
 $c_x, c_y$  – Principal point of the image  
 $k_n$  – Radial distortion coefficients  
 $p_n$  – Tangential distortion coefficients

The extrinsic parameters of the camera:

$X_c, Y_c, Z_c$  – Position of the camera  
 $\omega, \phi, \kappa$  – Rotation angles of camera



The geometric relationship between a point P in the space and its projection on the image plane is defined by their collinearity, i.e., the points have the property of lying on a single line. The collinearity equations are represented below involving the intrinsic and extrinsic parameters of the camera:

$$x = c_x - f_x \frac{r_{11}(X - X_C) + r_{12}(Y - Y_C) + r_{13}(Z - Z_C)}{r_{13}(X - X_C) + r_{23}(Y - Y_C) + r_{33}(Z - Z_C)}$$

$$y = c_y - f_y \frac{r_{12}(X - X_C) + r_{22}(Y - Y_C) + r_{23}(Z - Z_C)}{r_{13}(X - X_C) + r_{23}(Y - Y_C) + r_{33}(Z - Z_C)}$$

where:

- $r_{ik}$  are the elements of the rotation matrix  $R_{3 \times 3}$  defined by the rotation angles of the camera  $\omega, \phi, \kappa$  mentioned above
- $(X, Y, Z)$  are the three dimensional coordinates of a point P in the space in a Cartesian coordinate system
- $(x, y)$  are the bi-dimensional pixel coordinates of point P on the image, already corrected from lenses distortion.

The equations that relate distortion corrected and not corrected pixel coordinates of a point on the image can be found in [OpenCV].

## 8.2.4 Interpolating Camera Models

Most cameras have fixed geometry, thus a single camera model (pointed appropriately) applies. However, some cameras have variable geometry, meaning the camera model varies as a function of camera parameters.

The most obvious example of this on M2020 is the ZCAM, which has both variable zoom and focus. RMI and Watson also have variable focus. It is also possible to interpolate over temperature, which was used on MSL.

In order to deal with this, camera model interpolation is used. The camera is calibrated at a number of key points (in, say, focus or zoom space) and then the model is interpolated between those points as needed.

An interpolated model is created based on a piecewise linear interpolation of the model parameters. The calibration models are ordered based on the interpolation value, and the nearest two to the actual value are used. Note that if the variable is outside the range of the calibration models, the values are linearly extrapolated using the last two models.

Then, each of the C, A, H, V, O, and R vectors is independently linearly interpolated between the two nearest models. The derived parameters Hs, Hc, Vs, and Vc (horizontal and vertical scale and center, see [Yakimovsky1978]) are interpolated as well.

After interpolation, the A and O vectors are normalized (to keep them unit vectors). The H and V vectors are then reconstructed from the interpolated A, Hs, Hc, Vs, and Vc. This means that the interpolated H and V are not actually used.

Finally, the interpolated model is pointed as with any other camera model [Bell2017].

For Mastcam-Z, it is a little more complicated, due to the two dimensions. First, the models are interpolated for focus at the calibrated zooms on either side, using a function that relates best-focus distance at different zooms. Then, those two models are interpolated based on the zoom value.

### 8.3 Pointing a Camera Model

This section describes how the camera model is determined for the mast-mounted cameras (Navcam, Mastcam-Z, and RMI) given the mast joint angles, a process known as forward kinematics. It is adapted from Appendix A of [Bell-2017]. The calibration parameters (including calibration camera models) are all in the calibration collection, see Appendix C.

Note that the arm-mounted cameras do not go through forward kinematics; the FSW reports the final position and quaternion directly. Thus only the last section below applies.

Rotations are expressed using unit quaternions, which are four-tuples of the form

$$(s, v_x, v_y, v_z) = (s, \vec{V}) = \left( \cos \frac{\theta}{2}, \vec{A}_x \sin \frac{\theta}{2}, \vec{A}_y \sin \frac{\theta}{2}, \vec{A}_z \sin \frac{\theta}{2} \right)$$

where  $\vec{A}$  is the unit vector axis of rotation,  $\theta$  is the angle of rotation, and the magnitude (square root of the sum of each of the four elements squared) is 1. Multiplying two quaternions composite their rotations (rotate by  $Q_2$  first and then  $Q_1$ ):

$$Q_1 \times Q_2 = (s_1 s_2 - \vec{V}_1 \cdot \vec{V}_2, s_1 \vec{V}_2 + s_2 \vec{V}_1 + \vec{V}_1 \times \vec{V}_2)$$

which uses vector dot and cross products. Rotating a vector by a quaternion is then:

$$Q \times \vec{V} = \text{vector}(Q \times (0, \vec{V}) \times Q')$$

where the central term builds a quaternion with scalar 0 and vector component  $\vec{V}$ ,  $Q'$  is the quaternion inverse, which negates the vector component, and  $\text{vector}()$  returns the vector part  $\vec{V}$  of the quaternion (simply discarding the scalar).

The calibration camera models were taken at a specific pose, defined by the parameters “rsm\_calibration\_position” and “rsm\_calibration\_quaternion” in the M20\_mast.point calibration file. To construct the new pose P, Q, we extract the ARTICULATION\_DEVICE\_ANGLE from the RSM\_ARTICULATION\_STATE\_PARMs group in the VICAR/ODL label (or equivalent in the PDS4 label). The first two elements are the measured joint angles in the azimuth (az\_target) and elevation (el\_target) directions. These are *not* azimuth and elevation, but are rather angles from the hard stop in the two directions, which are nominally at azimuth 181 and elevation -91 degrees. Thus az\_target=181 is (approximately) straight forward along the +x rover axis, and el\_target=91 is (approximately) level.

The following pseudocode creates the rotation quaternion using az\_target, el\_target, and the boldface constants from the M20\_mast.point calibration file and epsilon = 1e-10.

```
az_axis = normalize(az_axis)           // to unit vector
el_axis = normalize(el_axis)
```

```

// Rotate around az_axis to align with home_az, home_el
Quaternion axis_rot = Rotation(az_axis, (az_home - el_azimuth))
Vector el_axis_rot = axis_rot × el_axis // Rotate a vector
Vector el_point_rot = axis_rot × (el_point - az_point) + az_point

// Transform target angles they're 0 at home
double az_angle = az_target - home_az
double el_angle = el_target - home_el

// Compute reference point position Pr at home configuration
Vector tmp1 = az_axis × el_axis_rot // Cross product
Vector Pr = el_point_rot
if (magnitude(tmp1) > epsilon) { // not parallel, safe
    double det = el_axis_rot · az_axis // dot product
    double alpha = (az_point - el_point_rot) · el_axis_rot +
        ((el_point_rot - az_point) · az_axis) * det
    alpha = alpha / (1.0 - det*det)
    Pr = el_point_rot + el_axis_rot * alpha
}
// Compute final point and rotation as composite of az and el
Quaternion az_rot = Rotation(az_axis, az_angle)
Vector P_rmech = az_rot × (Pr - az_point) + az_point
Vector P = P_rmech + rmech_to_rnav // Final point, RNAV
Quaternion el_rot = Rotation(el_axis_rot, el_angle)
Quaternion Q = az_rot × el_rot // Final quaternion

```

Note that the kinematics algorithm works with the Rover Mechanical frame (Section 20.3) throughout, converting to Rover Navigation frame (Section 0) at the end. The calibration and final camera models are expressed in Rover Navigation frame.

Finally, the camera model extrinsics must be transformed from  $P_{cal}$ ,  $Q_{cal}$ , to  $P$ ,  $Q$ . This part applies to arm-mounted cameras as well as mast-mounted cameras. To do this, we compute a composite transform that “unrotates” by  $Q_{cal}$  and then rotates by  $Q$ :

$$Q_{rot} = Q \times Q'_{cal}$$

The CAHVORE vectors are then transformed as follows:

$$\begin{aligned}
 C' &= Q_{rot} \times (C - P_{cal}) + P \\
 A' &= Q_{rot} \times A \\
 H' &= Q_{rot} \times H \\
 V' &= Q_{rot} \times V \\
 O' &= Q_{rot} \times O \\
 R' &= R \\
 E' &= E
 \end{aligned}$$

For full-frame images, this is the final result. Subframed or downsampled images must be additionally transformed.

The starting coordinate of a subframe is found in the `FIRST_LINE` and `FIRST_LINE_SAMPLE` VICAR/ODL keywords. `FIRST_LINE` defines the line start (dy) and `FIRST_LINE_SAMPLE` defines the sample start (dx). Note that the number of lines or samples does not affect the camera model. The model is then transformed as follows:

$$\begin{aligned} H' &= H - (dx - 1)A \\ V' &= V - (dy - 1)A \end{aligned}$$

The -1 is because the keywords start counting at 1, by definition. If there is no subframe, dx and dy are 1 and there is no modification. The other vectors are unchanged.

The downsampling factor is in the `PIXEL_AVERAGING_HEIGHT` (hscale) and `PIXEL_AVERAGING_WIDTH` (vscale) keywords. Downsampling is accomplished via the following:

$$\begin{aligned} H' &= H / hscale \\ V' &= V / vscale \end{aligned}$$

which should be done after the subframe shift. To be technically correct, one additional step is needed to compensate for integer coordinates being in the center of a pixel: the H and V vectors must be shifted by -0.5 pixel before the scale and +0.5 pixel afterward, making the full equations:

$$\begin{aligned} H' &= (H + A/2) / hscale - A/2 \\ V' &= (V + A/2) / vscale - A/2 \end{aligned}$$

## 8.4 Epipolar alignment and Linearization

EDRs and single-frame RDRs are described by a camera model. This model, represented by a set of vectors and numbers, permits a point in space to be traced into the image plane, and vice-versa.

EDR camera models are derived by acquiring images of calibration targets with known geometry at a fixed azimuth/elevation. The vectors representing the model are derived from analysis of these images. These vectors are then translated and rotated based on the actual pointing of the camera to represent the conditions of each specific image. The resulting vectors make up the “camera model” for the EDR. (see [DEEN1992], [YAKIMOVSKI1977], and [GENNERY]).

The Engineering Cameras use a general CAHVORE model, while all other cameras use CAHVOR. Both model types are nonlinear and involve some complex calculations to transform line/sample points in the image plane to XYZ positions in the scene. To simplify this, the images are “warped”, or reprojected, in a process often called “linearization”, such that they can be described by a linear CAHV model. This linearization process has several benefits:

- 1) It removes geometric distortions inherent in the camera instruments, with the result that straight lines in the scene are straight in the image.
- 2) It aligns the images for stereo viewing, known as epipolar alignment. Matching points are on the same image line in both left and right images, and both left and right models point in the same direction.
- 3) It facilitates correlation, allowing the use of 1-D correlators in some cases.
- 4) It simplifies the math involved in using the camera model.

However, linearization has some drawbacks (described below), which make it difficult to use for Mars 2020. For this reason, very little work is done with linearized images, a departure from prior missions.

The linearized CAHV camera model is derived from the input camera model by considering both the left and right eye models and constructing a pair of matched linear CAHV models that conform to the above criteria. For details on this algorithm see references [YAKIMOVSKI1977], and [GENNERY].

There are two types of linearization. When stereo is implemented via two cameras at a fixed distance from each other, as in most other Mars surface missions, “nominal” linearization can be used. The stereo partner can be predicted based on the fixed camera geometry, and this model can be used for linearization. This allows each image to be processed independently, without the need to find the stereo partner (or even acquire the image), yet provides the same results as if the match had been performed. The other type is “actual” linearization, where the actual partner image is used, rather than a prediction of what it should be.

Unfortunately, nominal linearization is not feasible for Mars 2020. For the Mastcam-Z, it is impossible to predict the zoom and focus of the partner. Even if the zoom were the same (as will often be the case, but not always), the autofocus works separately for each eye. For the Navcams and Hazcams, the tiled nature of the cameras means we cannot in general predict which tiles will be downlinked or even acquired. The geometry of the partner (both in terms of downsample and subframe) is thus not predictable. The rest of the cameras are not stereo. Although stereo can be performed with arm-mounted cameras, predicting the location of the partner is similarly impossible.

This means that all M2020 linearization must be done in “actual” mode. But actual mode has a number of drawbacks. The linearization itself is critically dependent on geometry, so if pointing correction is used, it’s not a simple matter of adjusting the pointing of the result – the entire linearization must be rerun. This makes creation of corrected mosaics and meshes extremely difficult. The tiled ECAMs may change in geometry as more tiles come in; this forces recomputation of data from *both* eyes. Finally, and most importantly, missing ECAM tiles force very difficult linearization geometries, which leads to extreme distortion of the images.

For these reasons, linearization is performed for the purposes of making anaglyphs (Section 8.14) but is not used otherwise – although linearized stereo products may be generated and archived.

## 8.5 Disparity

The parallax, or difference measured in pixels, between an object location in two individual images (typically the left and right images of a stereo pair) is called the “disparity”. Disparity files contain these disparity values in both the line and sample dimension for each pixel in the reference image. This reference image is traditionally the left image of a stereo pair, but could also be the right image. The geometry of the Disparity image is the same as the geometry of the reference image. This means that for any pixel in the reference image the disparity of the viewed point can be obtained from the same pixel location in the Disparity image.

There are two types of disparity. In the primary type (DF\* for first stage and DS\* for second stage), the values in the Disparity image are the 1-based coordinates of the corresponding point in the non-reference image. Thus, the coordinates in the reference image are the same as the *coordinates* in the Disparity image, and the matching coordinates in the stereo partner image are the *values* in the

Disparity image. Disparity values of 0.0 indicate no valid disparity exists, for example due to lack of overlap, correlation failure, or parallax occlusion. This value is reflected in the MISSING\_CONSTANT keyword. Such holes are rather common depending on the scene.

Disparity images can also be “delta” disparity (DD\*), which measures the relative offset between coordinates in the two images. This is what most imaging scientists mean by disparity. These products are not produced in the nominal pipeline, but can be produced as special products. Missing values are flagged by the value specified in MISSING\_CONSTANT, typically 0.0.

For each of primary types, several kinds of file can be produced: normal, line, sample, first-stage, raw, error metric, and grid. Not all kinds apply to all types.

Normal disparity files contain 2 bands of floating-point numbers in (line, sample) order using the Band Sequential format. The line and sample components may be (but rarely are) stored in separate single-band files. DSP is the primary normal disparity file, and what should be used for most purposes.

First-stage disparity files are normal disparity files in format, but contain the first stage of a two-stage correlation process. They should be considered an intermediate result.

Raw files contain the results before doing left->right and right->left reconciliation. They should be considered an intermediate result.

Disparity error metric files contain information about the quality of the correlation match. The tools to create them have not yet been developed; they are listed here as a placeholder for future expansion. Thus the format has not yet been determined as of this writing.

Grid overlays are an aid to visualization that may be created on occasion. These files are single-band byte images showing how a regular grid is distorted by the disparity matches (which is itself an indication of the terrain).

Disparity coefficients files (COE/COF) contain the transform coefficients for each pixel in the file. The second-stage correlator uses the following general transform to map the local geometry between the two images:

$$x' = ax + by + c + gxy$$

$$y' = dx + ey + f + hxy$$

The coefficients files are 6-band floating point files containing in order: a, b, d, e, g, h. Depending on the transform chosen in the processing (e.g., similarity, affine), unused coefficients are set to 0. The c and f terms are the translation terms and they are encoded in the actual disparity output (recall that the disparity file contains the matching coordinate, not just the transform coefficient). COF files are used to communicate the transform between the first and second stage correlators, while COE is the output of the second stage correlator. Both are optional and not always used.

## 8.6 Correlation

The disparity explained in Section 8.5 is computed from stereo pair image correlation. The construction of the disparity files is a 2-step process:

### First stage correlation:

Uses the camera models, i.e., internal and external camera information, to define a search area in the right image where a given pixel in the left image should be looked for. The correlator then searches the area to find the corresponding pixel.

On prior missions, the first stage correlation approach had been to “linearize” the images, that is, to remove distortions due to the camera lenses and project the images in a theoretical, distortion free, stereo setup geometry. In this setup, the images were epipolar-aligned, allowing for the search to be limited to one dimension (a pixel in the left image would have its corresponding pixel in the right image on the same line), allowing for faster processing and reducing the chances of wrong matches by limiting the search area. However, the warping distortion of the images induced by the linearization can be severe depending on the acquisition geometry. With nominal stereo setup (e.g., navcam, hazcam), it is usually limited and reasonable, but with more complex acquisitions, for instance close-range mast work, different positions (for an arm-based camera), different rover locations (long-baseline stereo), different conditions (such as different focus or zoom on the Mastcam), different tiling parameters (for ECAM), or even different instruments, the linearization distortion can impair subsequent correlation. This is why linearization is not used on M2020.

For this mission, the first stage correlation does not linearize the images. A new correlator implementation is used that uses the camera models directly to look for corresponding pixels directly in the raw image along the epipolar line (for CAHV camera) or curve (for CAHVOR/E camera). Rather than distorting the images so the epipolar curves are straight, this new correlator follows the epipolar curve in the raw image. The advantage compared to “linearization” is an increased robustness to non-traditional stereo acquisition geometries. This in turn enables co-registration as described in Section 15.1.

### Second stage correlation:

This is a refinement step. The second-stage correlator does not use camera models and is based solely on the first stage correlation output. For each reference pixel, it uses the starting point from the first stage correlation to find the approximate corresponding pixel in the second image. It then looks only in the vicinity of this approximate location to refine the location to a subpixel accuracy. In addition to the refinement, the second stage correlation tries to recover pixels that failed during the first stage by using neighboring successful pixels as a starting point. This approach can fill in contiguous areas, but cannot jump to new “islands” of discontinuous disparity (such as at the edge of a rock) that were not found by the first stage.

## 8.7 XYZ Generation

The pixels in an XYZ image are coordinates in 3-D space of the corresponding pixel in the reference image. This reference image is traditionally the left image of a stereo pair, but could be the right image as well. The geometry of the XYZ image is the same as the geometry of the reference image. This means that for any pixel in the reference image the 3-D position of the viewed point can be obtained from the same pixel location in the XYZ image. The 3-D points can be referenced to any of the M2020 coordinate systems (specified by DERIVED\_IMAGE\_PARMS Group in the PDS label).

Most XYZ images will contain "holes", or pixels for which no XYZ value exists. These are caused by many factors such as differences in overlap, correlation failures, and the failure of a result to meet quality checks in the XYZ program. Realize this list of factors is only representative, and not exhaustive. Holes are indicated by X, Y, and Z all having the same specific value. This value is defined by the MISSING\_CONSTANT keyword in the IMAGE object. For the XYZ RDR, this value is (0.0,0.0,0.0), meaning that all three bands must be zero (if only one or two bands are zero, that does not indicate missing data). Note that "0.0,0.0,0.0" is technically a legal value, but could occur at most once in an image and will rarely occur at all. The value is based on legacy software from previous missions and is compatible with current mission software. Additionally, it is extremely unlikely that the value will conflict with actual data since it is between the rover's wheels in Rover Frame (which cannot be imaged in stereo except for heroic SHERLOC-WATSON efforts); even when the Site origin is visible (e.g. the rover moves away), the possibility of sampling an exact value of "0.0,0.0,0.0" is considered extremely low. Also, if the value were to be sampled as actual data, losing a single pixel in the image is not problematic. The file format does not support nulls, so some other sentinel value would have to be chosen instead.

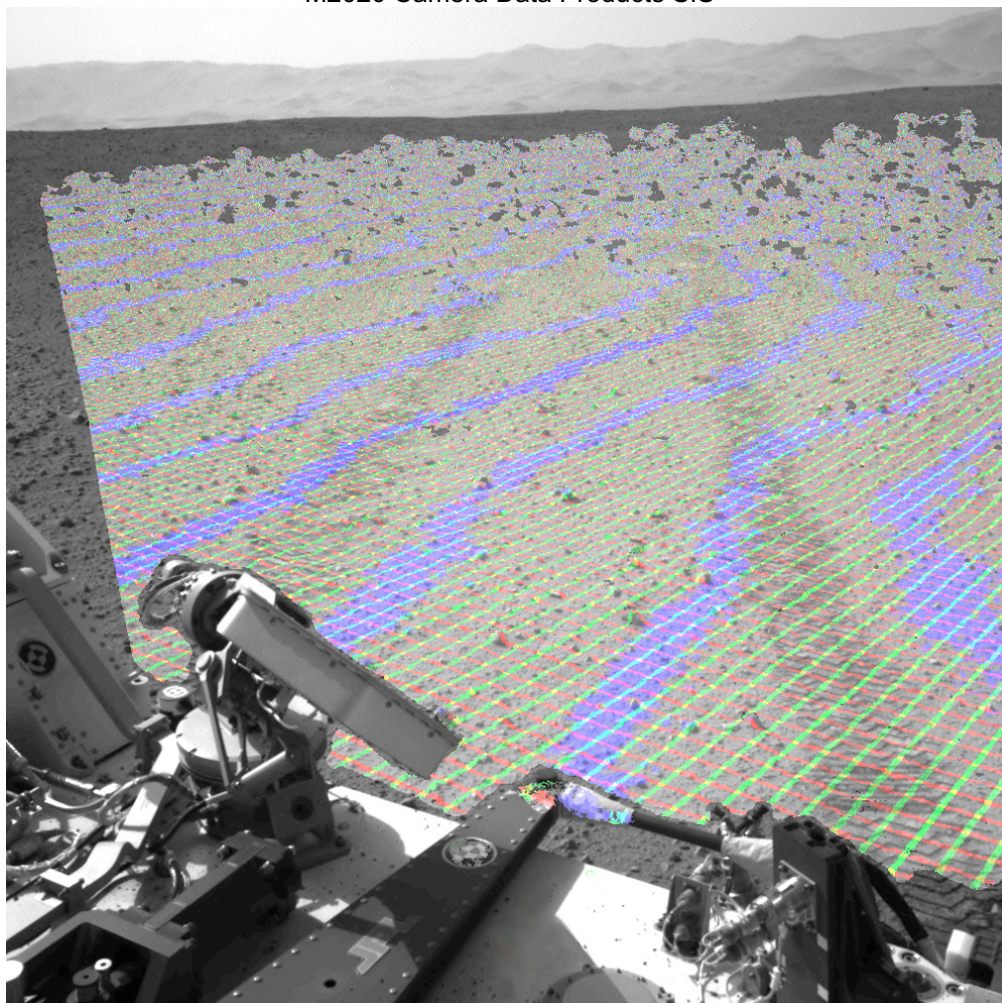
An XYZ Error metric (XYE) is available, which gives the estimated error for each pixel. It is a 3-band float product, with the three bands indicating the estimated range error (in meters) along each of the X, Y and Z axes. These values together define the error ellipsoid. Note that these values are axis-aligned, while the error is naturally range-aligned. Therefore the Range Error (RNE) product will generally be more accurate. XYE is provided as a convenience and its ellipsoid will always completely contain the RNE ellipsoid.

XYZ files can be filled (XYA). Individual X,Y,Z files can be filled as well. They can have associated rover mask files (MXY) as well as becoming masked (XYM) and saved as an overlay (XYO for XYZ, ZZO for Z-band), as shown in Figure 6-4 below. The rover mask files are discussed in more detail in Section 8.11.

Note that the XYZ generation process itself generates raw XYZs (XRZ, XRR). These are converted to true XYZs (XYZ, XYR) via range filtering (next section).

Figure 8-1 shows an example XYZ image overlaid on a background, from MSL. Red lines are lines of constant X, green of constant Y, and blue of constant Z. The blue lines are thus traditional elevation contours. Line spacing is 0.1m.





**Figure 8-1: Example XYZ image overlay from MSL**

### 8.7.1 Orbital XYZ

The pixels in an XOZ image are coordinates in 3-D space corresponding to the view from a reference image that intersects with an orbital elevation model using the reference image camera model. The reference image will be either a left or right NAVCAM image. The geometry of the XOZ image is the same geometry as the reference image. This means that for any pixel in the reference image the 3-D position of the viewed point can be obtained from the same pixel location in the XOZ image. The 3-D points can be referenced to any of the M2020 coordinate systems (specified by DERIVED\_IMAGE\_PARMS Group in the PDS label).

All pixels that intersect the orbital elevation model based on the camera pointing will contain a corresponding XOZ value, which should compromise all 'surface' pixels. 'Sky' pixels for which no ray intersection can occur with the elevation model will be assigned a no data value of -100000 in all bands. Due to Mars' curvature, there should never be a surface pixel intersection at greater than or equal to 100 km. The orbital elevation model has no holes, so no values should be missing for 'surface' pixels. However, depending on how far away the reference pixel is looking, multiple orbital elevation models may be employed to find a corresponding XOZ intersection as the highest resolution orbital data (1 m/pixel HiRISE DEMs) do not completely cover the area outside the landing ellipse. Reference images covering the Jezero crater rim, especially to the north, east, and south, will require intersection with lower resolution data

This document has been reviewed and determined not to contain export controlled technical data.

(CTX at 20 m/pixel or HRSC at 50 m/pixel). However, since resolution falls off with range from the rover, these distant (tens of kilometers) pixel intersections shouldn't result in any noticeable difference with between orbital elevation models. 'Sky' are indicated by all bands having the same specified value. This value is defined by the MISSING\_CONSTANT keyword in the IMAGE object. For the XOZ RDR, this value is (-100000.0,-100000.0,-100000.0), meaning that all three bands must have the same value.

## 8.8 Range Error

The theoretical range error for a stereo camera is:

$$e = \frac{r^2 i c}{b}$$

or alternatively,

$$e = \frac{r^2 c}{f b}$$

where  $r$  is the range from the camera (in meters),  $i$  is the instantaneous field of view in radians (ifov: angular dimension of a pixel, usually at the center of the camera),  $c$  is the correlation accuracy (in pixels),  $b$  is the baseline (in meters), and  $f$  is the focal length in pixels (focal length / pixel pitch). Correlation accuracy is very difficult to assess and there is no real consensus within the MIPL team. We use 0.25, 0.33, or 0.5 pixels at various times (quarter, third, or half a pixel accuracy). Accuracy is almost certainly better than a half a pixel, so that serves as a useful upper bound.

The parameters in Table 8-1 summarize the major stereo cameras (excluding rear hazcam). Figure 8-2 shows the range error as a function of range for these cameras out to 10m, while Figure 8-3 shows it to 75m.

**Table 8-1 – Range Error Parameters**

Parameter	Navcam	Front Hazcam	ZCAM (26mm)	ZCAM (110mm)
ifov	0.32 mrad	0.46 mrad	280 $\mu$ rad	66.7 $\mu$ rad
baseline	0.424 m	0.25 m	0.244 m	0.244 m
pixel pitch	6.4 $\mu$ m/pixel	6.4 $\mu$ m/pixel	7.4 $\mu$ m/pixel	7.4 $\mu$ m/pixel
focal length	19.2 mm	13.8 mm	26 mm	110 mm

## Range Filtering

Range filtering compensates for XYZ's having an error in the range direction that is often much greater than the cross-range resolution. This difference causes "spikes" in the XYZ data. These spikes are greatly reduced or eliminated by range filtering.

The stereo range error equation is given in Section 8.7.1. The size of a pixel, measured horizontally (cross-range), is:

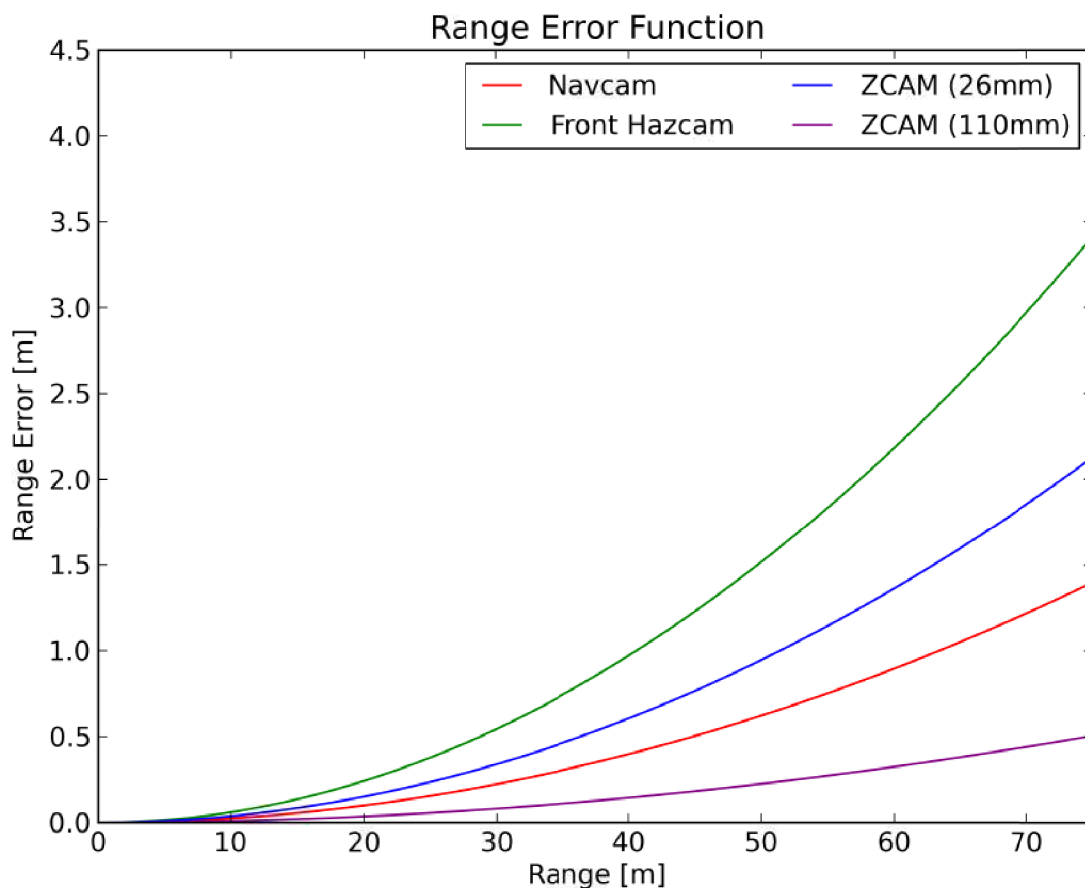
$$s = ri$$

Where  $r$  and  $i$  are defined as with range error. (Technically it's  $\tan(i)$  but the small angle approximation applies).

For most real stereo cameras, the stereo range error is much bigger than the cross-range pixel size. If we use the figures in Table 8-1 and assume a 0.25 correlation accuracy (a very aggressive assumption to minimize the error), we get the results shown in Table 8-2.

**Table 8-2 – Range Error vs Cross-Range Resolution (Mastcam-Z at 34mm)**

--	--	--	--	--	--	--



1 m	0.009 cm	0.0067 cm	1.35	0.0249cm	0.032 cm	0.778
5 m	0.225 cm	0.033 cm	6.76	0.623 cm	0.16 cm	3.89
10 m	0.902 cm	0.067 cm	13.52	2.49 cm	0.32 cm	7.78
50 m	22.6 cm	0.334 cm	67.62	62.3 cm	1.6 cm	38.92

While these are comparable close in, the ratio of error/size quickly gets quite large; up to 68 for the Mastcam-Z at 110mm zoom at 50m.

This ratio of range error to crossrange resolution is key. In practical terms, it means the noise in the along-range direction is e.g. 68 times the noise in the cross-range direction. This discrepancy leads to pronounced “spikes” in the XYZ data. These are most visible when looking at a terrain mesh. The farther out you go, the more prevalent these spikes (pointing back at the camera, or away from it) are. However, these spikes are pure noise, caused by the correlation uncertainty.

Range filtering significantly reduces this noise by converting the XYZ to range data (thus separating the noisy along-range data from the non-noisy cross-range data), filtering the range, and converting it back to XYZ. Filtering is based on the error formula, and is adaptive, so the proper amount of filtering is done at each point.

The result of this filtering is a trade-off between range error and spatial resolution of the range data. The filtered result is far less noisy (with less error), but the resolution of the range data is considerably less. You will not be able to see small terrain features (e.g. small rocks) after filtering, if the rocks are smaller than the filter size. But you are also not losing anything, because the range signal of that rock was swamped by the noise anyway.

Experiments show this method to be vastly superior to the raw XYZs for orthomosaic and mesh generation. Therefore, we use it throughout Mars 2020. The raw XYZ data is still available, as XRZ (site frame) or XRR (rover frame), if you want to use it.

### 8.9.1 Range Filter Algorithm Details

In a nutshell, range filtering implements a weighted plane fit to the range data around the given pixel, evaluating the plane at the given pixel. This has a similar effect to averaging, but gets around issues with biasing the results as the window slides off the edge of the image.

First the range from the camera is computed from the XYZ data. Then a preliminary lowpass filter (standard boxcar, excluding 0s) is performed on the range. This is used for determining the window size later. The low-pass filter prevents the window size from gyrating wildly from pixel to pixel, as it would do if we used the raw range. The filter does suffer from edge effects and other issues (which the main filter does not), but since the result is used only to size the window, small inaccuracies have a negligible effect.

The range filter itself proceeds as follows. Each pixel is computed independently. First, the nominal size of the filter half-window is determined. This is based on the ratio of error to size, and ends up being simply:

$$G_x = \frac{rc}{2b}$$

The correlation quality (c) is 0.25 so as to err on the side of less filtering. The range is the low-pass range discussed above.

The half-window height ( $G_y$ ) is half the width. This shrinks the kernel vertically, creating an ellipsoidal kernel, because there is typically much more range variation in the vertical direction than horizontal, for most *in-situ* scenes.

The program then loops over the window, computing a weight for each pixel. This weight is a combination of two factors: the Gaussian kernel, and a proximity filter. These two together implement a form of bilateral filter. The Gaussian kernel is:

$$w = e^{-\left(\frac{x^2}{2G_x^2} + \frac{y^2}{2G_y^2}\right)}$$

where x and y are the pixel coordinates within the window.

The proximity filter is used to reject values that are too far away in range space. This helps particularly when there are gaps due to occlusions. Without the proximity filter, the data on the far side of the occlusion (which could be tens of meters away, say due to a ridgeline) is included in the plane fit, which can bias the results, with the effect of the terrain “bending” to fill the gap. The proximity filter prevents this by de-weighting and eventually eliminating data that is too far away. The actual expected error in meters is determined using the actual central pixel. This error is then multiplied by the min (5) and max (10) parameters. The range of the pixel being considered in the window is then compared to the (actual) range of the central pixel. If the difference is within +/- of the minimum value (5 times the error), the Gaussian weight is passed through unchanged. If it is bigger than +/- the maximum value (10 times the error), the pixel is discarded (weight of 0). If it is in between, a proximity weight is applied to the Gaussian weight, which is linearly scaled from 1.0 at the min distance to 0.0 at the max distance. This has the effect of gradually rolling off the proximity window, which helps to avoid discontinuities in the data.

Once the final weight is determined for each pixel in the window, a plane fit is performed. The plane equation is:

$$z = Ax + By + C$$

and the solution matrix equation is:

$$\begin{bmatrix} \sum wx^2 & \sum wxy & \sum wx \\ \sum wxy & \sum wy^2 & \sum wy \\ \sum wx & \sum wy & \sum w \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum wxz \\ \sum wyz \\ \sum wz \end{bmatrix}$$

where w is the proximity-modified Gaussian weight, x and y are again coordinates within the window, and z are the range values. The matrix is solved using Cramer’s rule, and then evaluated at the center (x=0, y=0) to get the final range (meaning only C is relevant).

This final range value is then projected back into XYZ using the camera model.

It should be noted that there is one potential bias that is not accounted for here. As you get closer to the horizon in a typical *in-situ* scene, pixels get farther and farther apart, as the tangent of the view angle. This means that typically, pixels in the window below the central pixel will be closer to the central pixel in range space than pixels an equal number of lines above the central pixel. This will have the effect of biasing the results slightly away from the camera. However, this is a second-order effect that is not expected to be an issue for real images. It is partially mitigated for two reasons: the proximity filter will be more likely to cut off the distant pixels, and correlation tends not to work well when the view angle is close to tangent anyway.

The algorithm described above is referred to as "Range\_Filter\_Deen\_2020" in the PDS4 labels.

## 8.10 Range

Range is simply the Cartesian distance from a reference point to the XYZ coordinates of the pixel. Although the reference point can be arbitrary, only two are commonly used. RNG products describe the range from the camera (technically, the C point of the camera model). RNR products describe the range from the origin of the rover coordinate system (a point on the nominal ground, midway between the center wheels).

## 8.11 Rover Mask

The rover mask (MXY) file is used to mask off the rover, so that XYZ points correlated on the rover do not show up in terrains. This prevents the problem of e.g. a wheel being interpreted as real terrain, which would appear as an obstacle to driving. The rover mask is applied to the XYZ file to create a masked XYZ (XYM) file, which is just XYZ with points near the rover removed.

The rover mask is created using a medium-fidelity volumetric model of the rover, with some margin (default 1.5cm) around the rover. This model is articulated based on the telemetered joint positions of the arm, wheels, and suspension. The articulated model is then projected into the image to create a mask. The articulation means the mask will reflect the arm and wheel positions, minimizing the amount of good terrain that is masked off. If joint positions are not available, a "swept volume" is used, meaning the mask covers all possible positions of the articulating device. The HGA is always modeled as a swept volume, because it moves continuously and is unlikely to be in the same place as when the image was taken.

Rover masks can also be generated using predicted arm joint positions; this is the basis of the "SuperCam Finder" mosaics produced during operations. Such predicted-state masks are marked using the special processing flag field in the filename.

Associated with the MXY mask image is an XML file with the same base name, but an extension ".xmlf". This file contains a polygonal representation of the articulated volumetric model that was used to make the rover mask (before it was projected into the image). The format of this file is not described completely here, but it is straightforward XML, consisting of a large number of polygons in XYZ space. The ".xmlf" file is delivered to PDS as a "ancillary" text file, using a minimal PDS4 label.

## 8.12 Surface Normal

The pixels in a UVW image correspond to the pixels in an XYZ file, with the same image geometry. However, the pixels are interpreted as a unit vector representing the normal to the surface at the point represented by the pixel. U contains the X component of the vector, V the Y component, and W the Z component. The vector is defined to point out of the surface (e.g. upwards for a flat ground). The unit vector can be referenced to any of the M2020 coordinate systems (specified by the DERIVED\_IMAGE\_PARAMS Group in the PDS label).

Most UVW images will contain "holes", or pixels for which no UVW value exists. These are caused by many factors such as differences in overlap, correlation failures, and insufficient neighbors to compute a surface normal. Holes are indicated by U, V, and W all having the same specific value, defined by MISSING\_CONSTANT as (0.0,0.0,0.0).

Two special kinds of surface normal products are defined based on MER experience. These are not expected to be used during nominal M2020 ops, but are included in case they become needed. The UVP type projects the surface normal onto an arbitrary plane in space, so the unit vectors always lie parallel to the plane. The UVT type provides the angle between the surface normal and the same arbitrary plane in a single band.

The UVW product contains the surface normal resulting from analyzing a small patch of pixels, which is appropriate for arm work. In contrast, the UVS product contains the surface normal resulting from analyzing a much larger patch of pixels (comparable to the size of the rover), which is appropriate for driving slope determination. Specific patch sizes are operational tuning parameters which have not yet been set, but will be provided in the history portion of the embedded VICAR label.

## 8.13 Slope Maps

The Slope-related maps represent aspects of the slope of the terrain as determined by stereo imaging. The Slope Map is derived from surface normal generated using rover-sized plane fit. There are several slope types, each of which can additionally be overlaid on a background.

In the equations below,  $u$ ,  $v$ , and  $w$  are surface normal components, while  $x$ ,  $y$ , and  $z$  are components of cartesian coordinate system.

### 8.13.1 Slope (nominal)

Slope in degrees for each pixel.

$$slope = \frac{180}{\pi} \left( \frac{\pi}{2} + \tan^{-1} \left( \frac{w}{\sqrt{u^2 + v^2}} \right) \right)$$

### 8.13.2 Slope Heading

Magnitude of the normal unit vector projected onto the horizontal plane. It is directly related to  $\sin(\text{slope})$ .

$$slope \text{ heading} = \frac{180}{\pi} \tan^{-1} \left( \frac{v}{u} \right)$$



### 8.13.3 Slope Rover Direction

The component of the slope (in degrees) that was facing the rover, i.e. if the rover went radially outward from its current position, this indicates the climb or descent. In the formula below,  $R$  is the rover's position.

$$V = \frac{[x - R_x \quad y - R_y]}{\sqrt{(x - R_x)^2 + (y - R_y)^2}}$$

$$\text{slope rover direction} = -\frac{180}{\pi} \tan^{-1} \left( \frac{V_x u + V_y v}{-w} \right)$$

### 8.13.4 Slope Magnitude

Contains the magnitude of the normal unit vector projected onto the horizontal plane. It is directly related to  $\sin(\text{slope})$ .

$$\text{slope magnitude} = \sqrt{u^2 + v^2}$$

### 8.13.5 Slope Northerly Tilt

Contains the component of the slope in degrees that points north.

$$\text{northerly tilt} = \frac{180}{\pi} \sin^{-1}(u)$$

## 8.14 Anaglyph/Colorglyph

In order to view true stereo images, special hardware is required. Because this hardware is somewhat rare, anaglyphs and colorglyphs are created for some products, which can be viewed with standard red/blue or red/cyan glasses (the red lens should be on the left eye).

Anaglyphs are used for grayscale images, and do not convey a sense of color. The left image is used for the red band, and the right image is used for the green and blue bands (both green and blue because there is significant leakage of green pixels into most blue glasses lenses, and this helps compensate for the relatively poorer blue sensitivity of most people).

Colorglyphs are used for color stereo images. They convey an impression of color, by assigning the red band of the left eye to the red channel, and the green and blue bands of the right eye to the green and blue channels. Thus in areas where there is no stereo disparity, the color appears correct when viewed without glasses. When viewed with glasses, an impression of color is seen. Note that colorglyphs do not work well with saturated red or blue images, but such are rare in natural Martian scenes.

Both anaglyph and colorglyph use linearized images (Section 8.4) to achieve epipolar alignment, which means that matching features appear on the same image line in both eyes (i.e. disparity is horizontal only). This is critical for human viewing of stereo images; more than a few pixels of vertical disparity is hard for most people to view in stereo. This is one of the few uses of linearized images in Mars 2020 (unlike prior missions where they were ubiquitous).



## 8.15 Pixel Scale for Single-Frame Images

A common question that arises is, how does one determine the pixel scale (mm, cm etc) in an image? There is no one number that works across most images. Pixel scale is a function of range, which varies across images, sometimes widely. However, pixel scale can be determined at given points in the image. If the image is looking square-on to an object (often the case for arm cameras), the scale may be close enough across the image (i.e. same range) that a single number suffices. It is up to the user of the data to determine what tolerance for variation they'll accept. Note that the scale can also vary across the image due to camera effects.

Pixel scale is a function of the range to target in meters and the ifov in radians (instantaneous field of view, or angular resolution) of the camera:

$$scale = r * \tan (ifov)$$

The small-angle approximation for tangent applies, so for all practical purposes this is just range times ifov.

The ifov, or angular resolution, for the Mars 2020 cameras can be found in the camera characteristic tables in Section 2. However, it is important to realize that this is a representative number, either the value at the center or an average. The value can change across the image. This effect is small for cameras with a small FOV such as RMI or MCC, but can be quite large for fisheye cameras such as the Hazcams or Navcam.

For most uses of pixel scale, the representative number will suffice. However, a more accurate calculation can be done using the camera model. Project the pixel in question into space using the camera model, resulting in a view ray (vector pointing in the direction of the object for that pixel). Then move over one pixel and do the same. The angle subtended by those two view rays is the ifov for that pixel. If the view rays are unit vectors, the angle is the arc cosine of the dot product of the vectors.

Range can be more difficult to obtain. For some images, it's easy: find the RNG product (Range from Camera) and get the value at that point. However, not all images are in stereo, and not all pixels successfully correlate to get the range.

For these cases, there are a few options. The best one is to find the same feature in another image which does have range and use the range from that. For Navcam, Mastcam-Z, and RMI, the difference in range between the cameras is negligible for this purpose, so the range can be used directly. For arm-mounted cameras, the difference can be significant. For these, take the XYZ coordinate of the feature in the stereo pair, and compute the Cartesian distance between that and the C point of the camera model.

If there is no suitable stereo pair covering the image, the only remaining option is to assume a surface model, and intersect the view ray from the camera with that. That provides an XYZ point, so again take the Cartesian distance to the C point of the camera model. Mosaics (other than orthorectified) use a surface model, usually just a flat plane either level with the horizon or level with the rover. Another option would be to use an orbital surface mesh to find the surface. Using an assumed surface model is significantly less accurate than the other methods.

## 9. 3D SURFACE RECONSTRUCTION AND MESHES

Terrain generation methods described below are used for tactical operations and science planning and are essential to better understanding of the rover's surroundings and planning rover traverse, robotic arm placement and remote sensing activities.

### 9.1 Tactical Terrain Mesh

The terrain models are generated by triangulating XYZ point clouds using volume based surface extraction [Olson2007]. The original image is used as a texture map to add detail and color to the polygonal surface representation. Each point cloud corresponds to a stereo pair of images and triangulation produces a wedge of terrain model with the narrow end of the wedge pointing toward the cameras. Most XYZ images will contain holes, or pixels for which no XYZ values exist. Some of the factors that cause holes are parallax, occlusions, correlation failures, and failure of a result to meet quality checks of the XYZ computation program. While there are mesh generation parameters that will result in generation of polygons covering small holes, the intent is to preserve geometric structure of the input point cloud. SITE frames are used almost exclusively for all terrain products.

Digital Elevation Models (DEMs) are also produced as separate image products and are commonly referred to as height maps. Height Maps usually contain data corresponding to XYZ values, as well as interpolated data that fills the holes while preserving all measured data.

#### 9.1.1 Adaptive Mesh Polygon Sizing

Full-resolution meshes are huge, and loading more than a few of them is beyond the capabilities of most current programs. Additionally, and more importantly, full-resolution meshes exceed the actual resolution of the data once range filtering (see Section 8.9) is applied. Range filtering trades off cross-range resolution for downrange noise reduction, so the actual cross-range resolution is significantly less than might be expected based on the raw pixel resolution.

For these reasons, an adaptive triangle size algorithm was implemented for the mesh. This algorithm downsamples the mesh by various amounts depending on the range. Downsampling is only done by powers of 2, but is different for the line and sample directions. The sample direction corresponds to cross-range in most cases and is generally downsampled more than the line direction, due to foreshortening in the standard *in-situ* viewing geometry.

#### 9.1.2 Format Description

Terrain models are stored in Open Inventor binary format [OpenInventor]. The main benefits of this format are support for multiple Levels Of Detail (LOD), triangle striping for efficient rendering, and ease of assembling individual pieces into unified models via referencing. Texture Maps are stored as PNG 8-bit image files and are referenced by the Open Inventor files. Height Maps (DEMs) are stored in VICAR format image files and consist of 3 bands. Band 1 contains actual elevation data, band 2 fills areas for which there is no actual data using interpolation, band 3 provides a metric of how close a pixel value is to the actual data.

Meshes are also produced in OBJ format (see [OBJ]), and are used both for operations and for archiving to PDS. From the PDS standpoint, the OBJ file is a collection of tab-separated tables. Each of the major sections (vertices list, texture mapping cords, and the list of faces) are considered

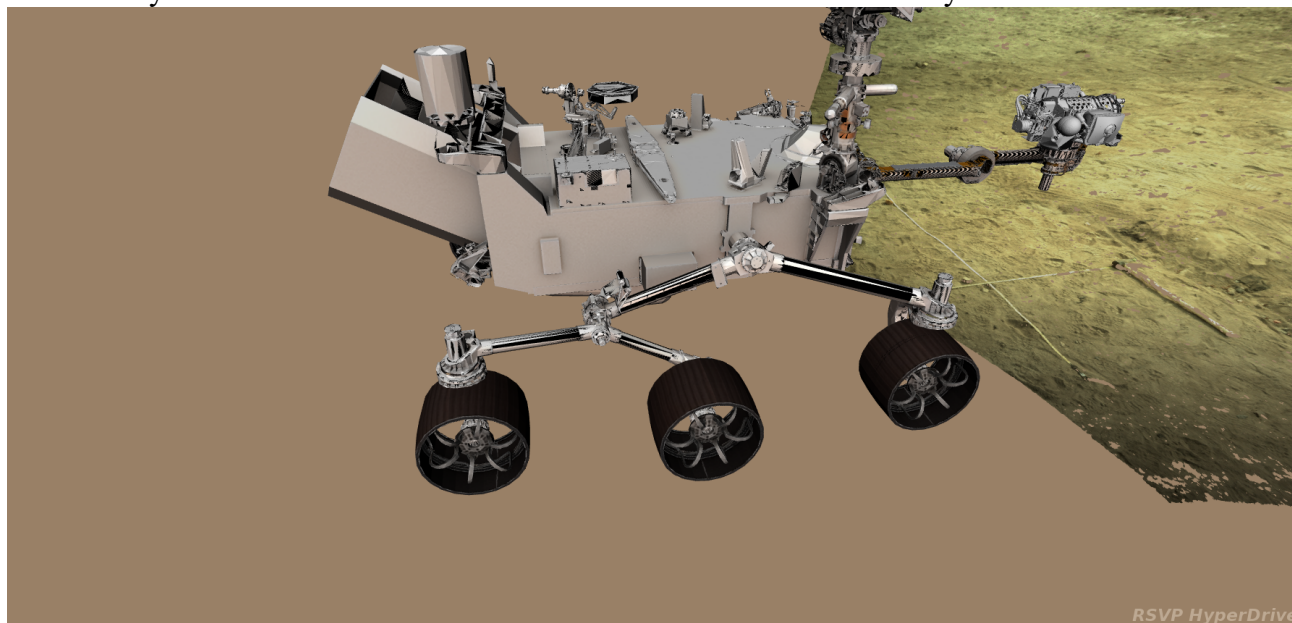
This document has been reviewed and determined not to contain export controlled technical data.

as a table with three or four elements (the type code and the parameters) and are described as such in the PDS label.

Note that it is perfectly valid to have an empty mesh (no triangles), for example if correlation completely failed. These empty meshes exist in the operations data set on the ODS, to indicate the file was processed but had no valid triangles. However, for PDS, these empty meshes are suppressed for technical reasons (PDS does not like 0-length tables) and so do not appear in PDS.

### 9.1.3 Unified Tactical Terrain Mesh

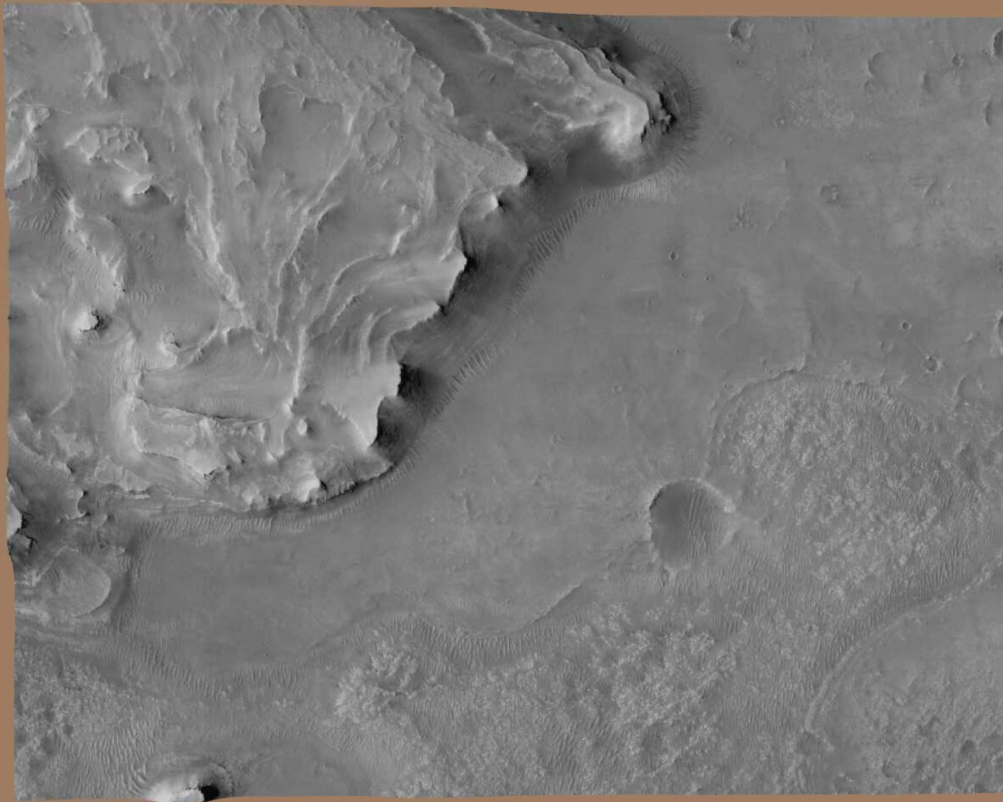
A separate process combines multiple wedges into a unified terrain model and corresponding unified height map. As new wedges become available, these individual terrain “pieces” are combined with existing terrains via a simple grouping. The complexity of resolving ambiguities while working with multiple terrains that can overlap and are not registered is left to the operations software suites. The advantage of this approach is that end-users are able to work with new wedges as well as existing terrain models within minutes of source data being downlinked from the spacecraft. Another benefit is the ability to evaluate terrain models from different sources individually.



**Figure 9-1: Robotic Arm workspace is rendered in Tactical Terrain Mesh and visualized in RSVP HyperDrive**

## 9.2 Orbital DEM-based Mesh

In addition to using in-situ image sources for modeling, terrain models generated from orbital MRO HiRISE camera datasets are regularly used for tactical sequencing of most drives longer than few meters as well as for strategic traverse planning over hundreds of meters. These terrain meshes (see Figure 9-2: Orbital terrain mesh of M2020 Landing Site in Jezero crater as visualized in RSVP Hyperdrive. are based on DEMs provided by the HiRISE team. The rover’s position in orbital maps is estimated by the project’s Localization Scientists and orbital terrain models that are generated with data expressed in the rover’s current SITE frame. It should be noted that new HiRISE images of the area are rare; the updated models use the same terrain information but are relocalized to the rover’s location each day.



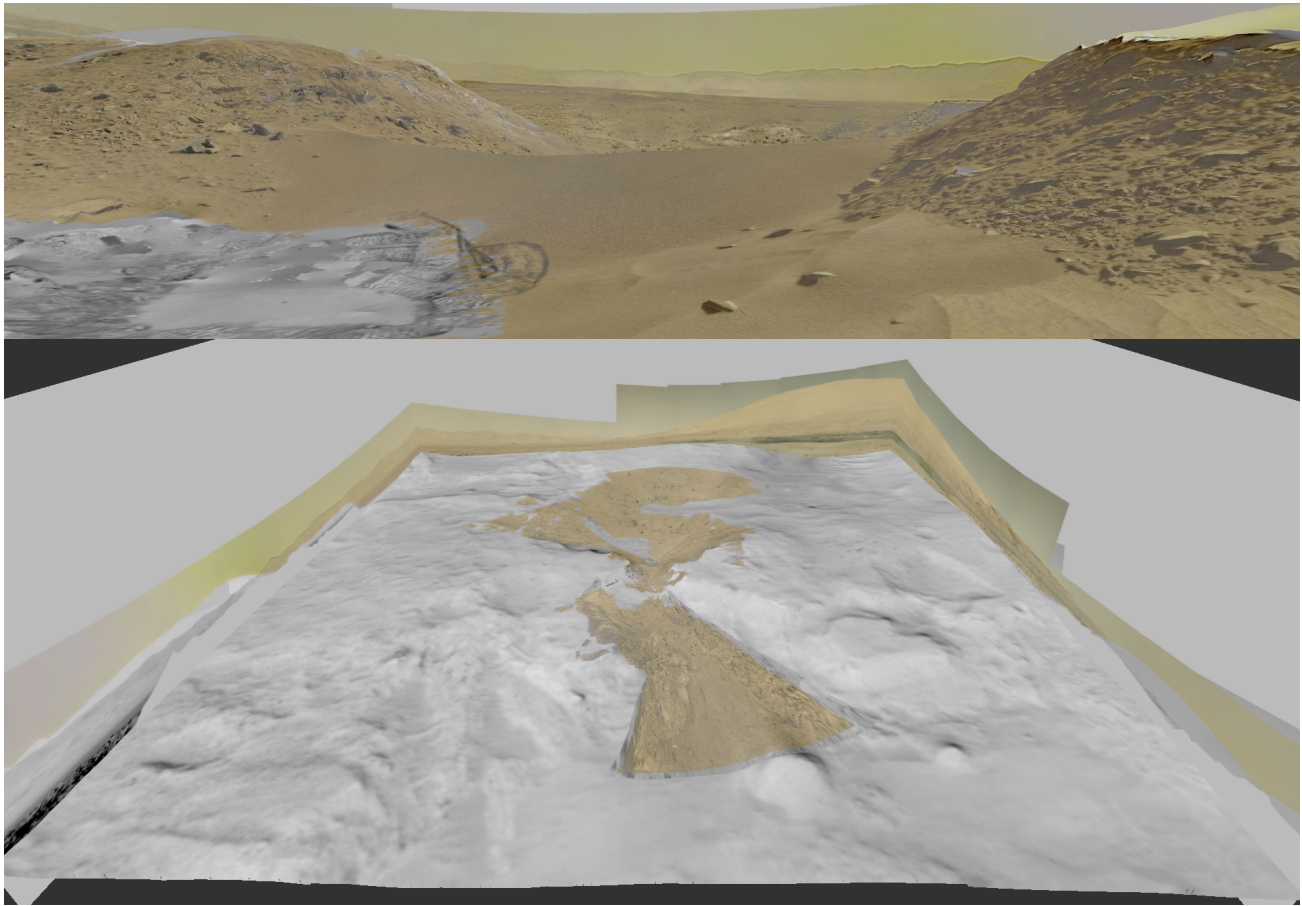
RSVP HyperDrive

Digital Elevation Models with 1M Resolution.

## Contextual Meshes

Whereas the per-wedge, unified, and orbital meshes are reconstructed from individual observations, contextual meshes fuse data from multiple observations. One contextual mesh is typically produced for each position of the rover during a given sol, and may combine geometry and imagery from up to thousands of individual in-situ and orbital observations. The combined observations can include data from multiple sols, multiple rover positions, and multiple instruments.

The intention of the contextual mesh is to provide spatial awareness. It can be visualized from a first-person navigable 3D point of view, showing not only local terrain features such as sand, pebbles, rocks, ridges, and hills, but also distant landmarks on the horizon and the skyline. It can also be viewed from a zoomed-out third person perspective, similar to looking at a diorama. Figure 9-3 shows an example of a contextual mesh.



**Figure 9-3: Contextual mesh in first person perspective (top) and third person perspective (bottom).**

Production of contextual meshes is typically fully automated and includes the following phases:

1. **Triggering:** possible creation of a new contextual mesh is triggered when new in-situ XYZ data becomes available for a rover position. The spatial extent of the contextual mesh is typically e.g. a 32 or 64m square in-situ region centered on that rover position, with an additional square orbital periphery extending to e.g. 64 or 256m.
2. **Input Collection:** heuristics are applied to search for available in-situ observations, including ones from other rover positions and other sols, which overlap the spatial extent of the contextual mesh. Applicable orbital observations are also gathered. The contextual mesh is only generated if sufficient data is available according to pre-set thresholds.
3. **Ingestion:** the inputs are filtered (e.g. to remove outdated versions) and processed to collect metadata including camera models and coordinate frames. This metadata is sourced from observation product headers as well as mission specific databases such as PLACES.
4. **Alignment:** starting with pose priors from the ingestion step, the relative poses of the observations are adjusted to compensate for alignment errors which arise from uncertainty in rover and instrument localization. Several algorithms are typically applied which automatically detect, match, and align geometric and visual features across observations.

5. **Implicit Surface Reconstruction:** Poisson surface reconstruction [Kazhdan2006] is applied to the aligned XYZ data to reconstruct a triangle mesh representing the in-situ data. This in-situ mesh is then blended with any orbital data using mesh processing algorithms.
6. **Subdivision:** The large terrain mesh (up to millions of triangles) from the reconstruction step is adaptively subdivided into smaller chunks, called leaf tiles, for further processing.
7. **Texturing:** a visual texture image is composed for each leaf tile by backprojecting data from the best available in-situ observation that observed each point in a fine sampling of the tile mesh. Points not observed in-situ may be textured from orbital imagery, if available.
8. **Exposure Seam Correction:** even when radiometrically calibrated images are used, typically there are still variations in brightness between adjacent observations, resulting in visible artifacts. A multigrid-based exposure seam correction algorithm [Kazhdan2010] is applied to adjust texture pixels across all leaf tiles to reduce these artifacts.
9. **Hierarchical Levels of Detail:** to facilitate streaming and portable visualization, coarser levels of detail are created as a hierarchical tree of parent tiles. Floating scale surface reconstruction [Fuhrmann2014] is used to generate parent tile meshes. The combined tree of parent and leaf tiles is saved in the 3DTiles format [3DTiles] for later visualization.
10. **Skybox:** views of distant features, the horizon, and the sky can be crucial for spatial awareness. A skybox is created that surrounds the contextual terrain mesh. It is textured, exposure seam corrected, and saved in the 3DTiles format using algorithms similar to those used for the terrain mesh.
11. **Scene Manifest:** a file is generated that lists all observations used in creation of the contextual mesh, the results of the alignment algorithms, related metadata, and the names of the output 3DTiles tilesets.

Like the products of other data fusion and reconstruction algorithms, e.g. computed tomography, and considering that it is produced automatically from inputs with noise and outliers, the contextual mesh may contain some artifacts. For example, the boundaries between areas reconstructed primarily from surface data and those reconstructed from orbital data may have some discontinuities. Outlier images with contrast or brightness extremes may affect regions of the sky. Contextual spatial understanding can generally still be gained even in the presence of such artifacts, though critical or quantitative uses can and should be cross-checked with other products.

The contextual mesh system differs from many other photogrammetry and scene reconstruction systems in that it heavily leverages properties of planetary surface mission datasets, such as the availability of typically good pose priors, stereo vision, calibrated cameras, and orbital data, to enable relatively fast and completely automated processing without manual intervention.



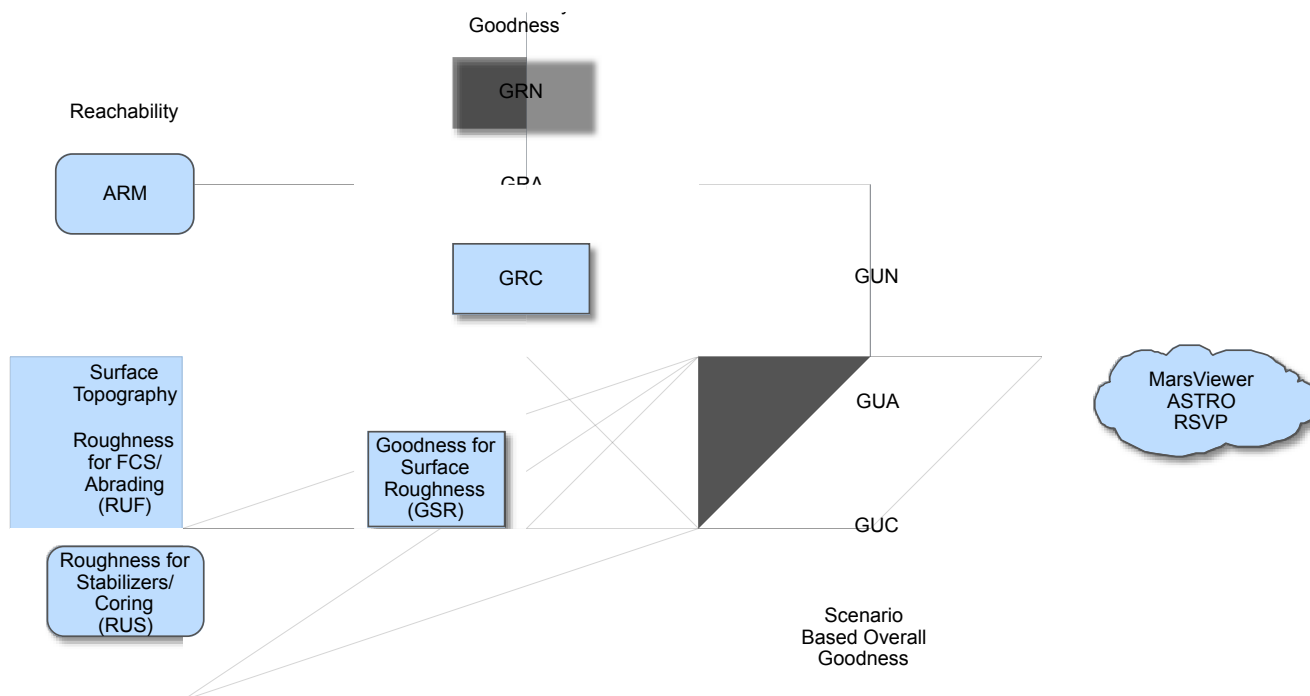
## 10. CONTACT SCIENCE PROCESSING

The primary purpose of Mars 2020 is to obtain samples for eventual return to Earth. This section describes the products used to assess whether or not a given target is amenable to sampling.

There are three “scenarios” for contact science analysis:

- 1) Natural Surface: is the surface suitable for examination with various instruments, without modifying it
- 2) Abraded Surface: is the surface suitable for use with the abrading bit
- 3) Coring: is the surface suitable for sampling (coring) with the drill

There are two types of primary analysis products: surface roughness and arm reachability. These are then analyzed in “goodness” maps, which indicate whether the surface meets requirements for a given scenario at each point. The goodness maps are the starting point; they provide a simple go/no-go indication across the entire image. The other products serve as backup material; if the goodness map shows no-go, the other products can be used to determine why.



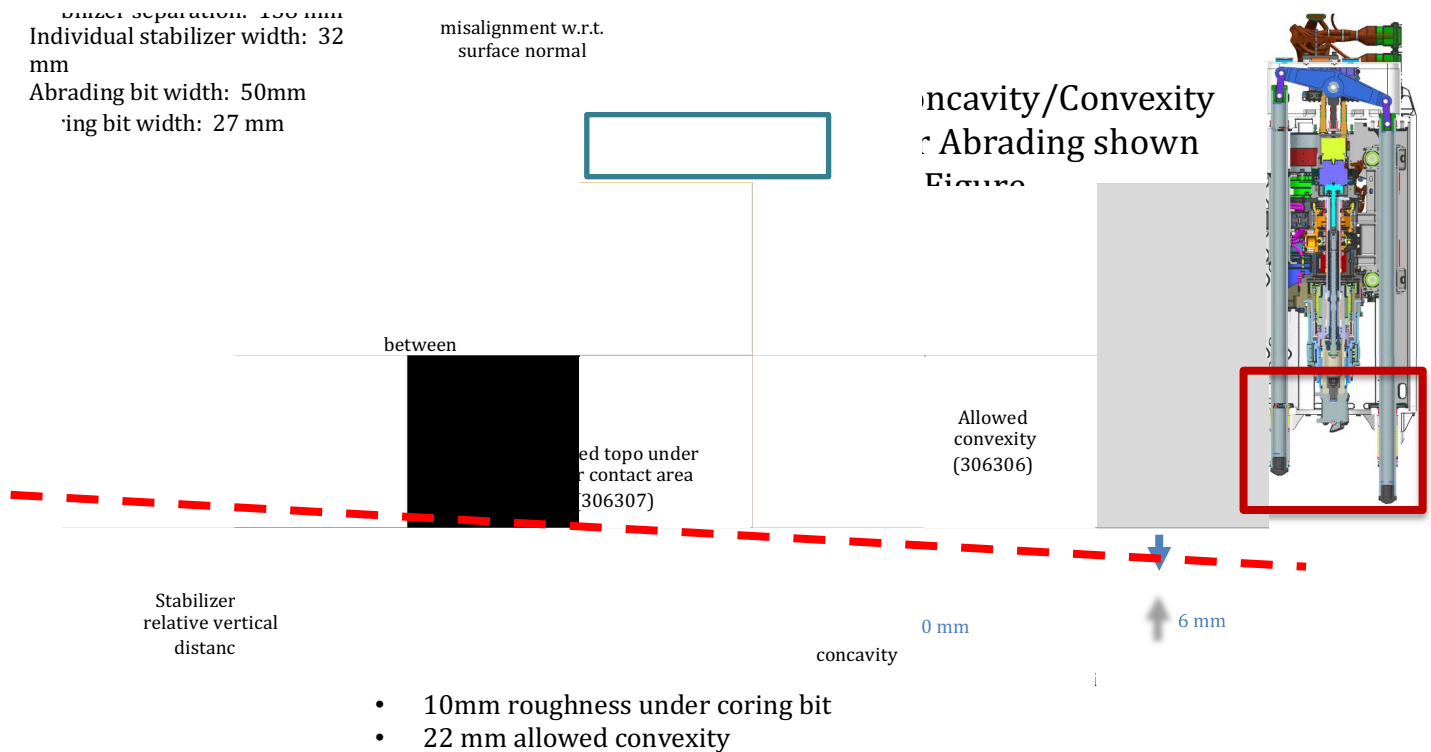
**Figure 10-1: Contact Science Product Flow**

## Surface Roughness

The roughness maps contain surface roughness estimates at each pixel in the image, along with a "goodness" flag indicating whether the roughness meets certain criteria.

For each pixel, the surface normal defines a reference plane. XYZ pixels in the area of interest are gathered, and the distance to the plane is computed. Outliers are thrown out. For the remainders, the minimum and maximum distances from the plane are found. Roughness is defined as the distance between this min and max (thus, is peak-to-peak variation within the area along the normal vector).

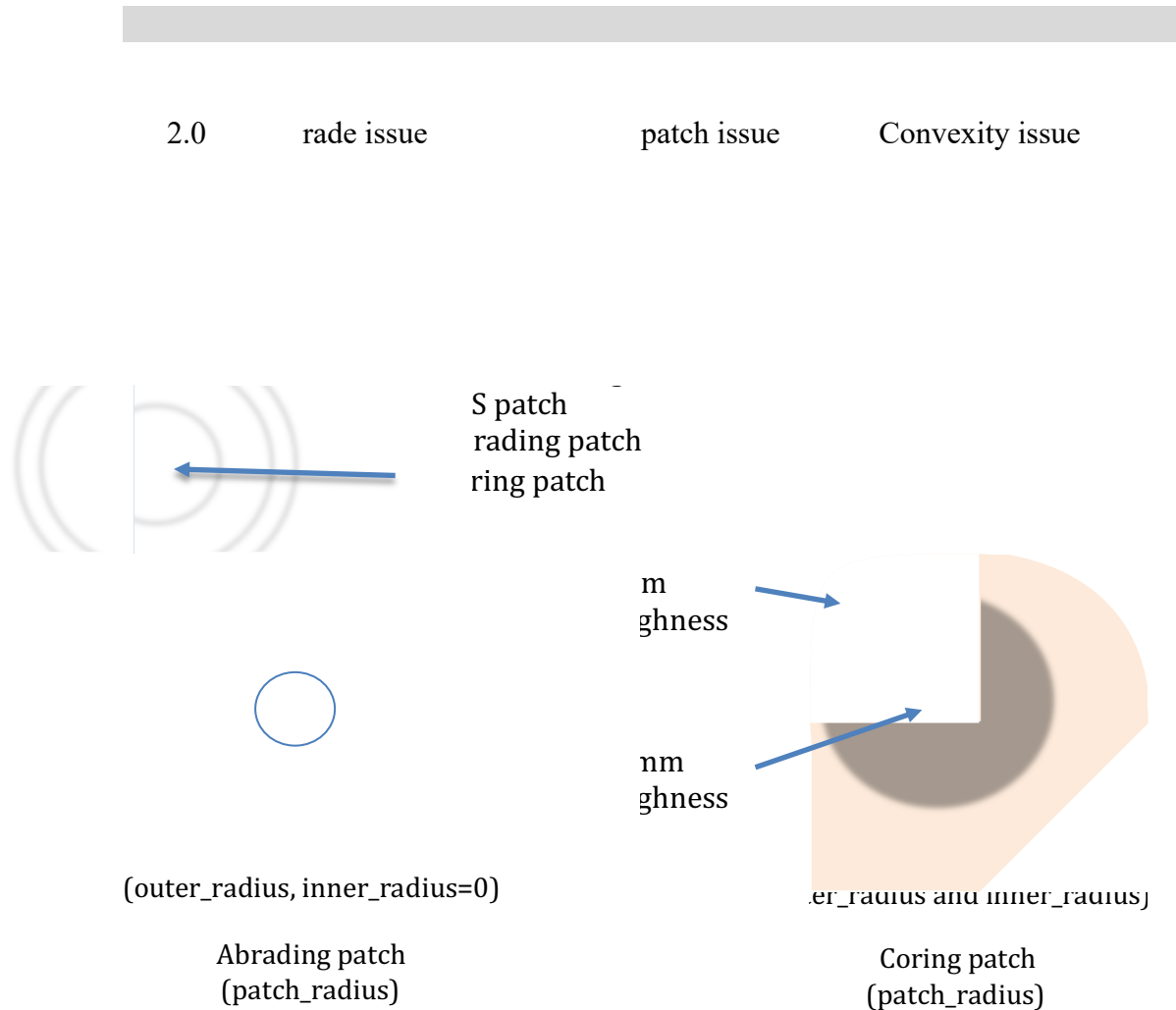
Surface Curvature is defined slightly differently. Convexity is the distance from the plane defined by the stabilizer bars to the maximum extent of the terrain above that plane. Similarly, concavity is the distance from that plane to the minimum extent of the terrain below that plane. Figure 10-2 shows the meaning of these products as applied to the drill.



The roughness products are defined over several sizes based on the drill requirements; Figure 10-3 shows this. There is a "ring" (which in the case of RUF extends to the center), and a "patch".

In all cases, the computed roughnesses or curvatures are compared to thresholds, which determine whether the point is "good" or not. A status band indicates this, as shown in Table 10-1. For RUC, the curvature values are evaluated separately for the Abrasion and Coring scenarios.



**Table 10-1 – Roughness Status Band Values**

## Arm Reachability

The Arm Reachability maps contain information about whether or not the instruments on the Arm can "reach" (contact or image) the object or location represented by each pixel in the scene. They are derived from the XYZ and Surface Normal (UVW) products.

The geometry of the reachability maps match the reference XYZ, and Surface Normal (UVW) images, in that each pixel in the file directly corresponds to the pixel at the same location in the other products.

The reachability map (ARM) encodes information for each of the 6 arm instruments in each of the 8 possible arm configurations, for a total 48 values per pixel. It is stored as a 6-band image of 16-bit integers in standard Band Sequential order. Each band represents one of the 6 arm instruments in the order defined by INSTRUMENT\_BAND\_ID (see Table 10-2). Within each band, the 16-bit integer contains 2 bits of information for each of the 8 configurations (in the order defined by

CONFIGURATION\_BIT\_ID) packed into the 16-bit value (see Table 10-3). The first mode in CONFIGURATION\_BIT\_ID is in the high-order 2 bits of the integer. The two bits represent 4 states, as indicated in Table 10-4. IK here means Inverse Kinematics, which indicates whether the point is physically reachable, independent of terrain. Collision indicates the arm would collide with the terrain.

**Table 10-2 – Arm Instruments**

Band	Instrument	Natural	Abraded	Coring
1	DRILL		X	X
2	GDRT		X	
3	SHERLOC WATSON	X	X	X
4	SHERLOC	X	X	X
5	PIXL	X	X	X
6	FCS	X	X	

**Table 10-3 – Arm reachability bit assignments per configuration**

16-bit Integer Bit Order (15=MSB, 0=LSB)							
15 - 14	13 - 12	11 - 10	9 - 8	7 - 6	5 - 4	3 - 2	1 - 0
Shoulder Out, Elbow Up, Wrist Up	Shoulder Out, Elbow Up, Wrist Down	Shoulder Out, Elbow Down, Wrist Up	Shoulder Out, Elbow Down, Wrist Down	Shoulder In, Elbow Up, Wrist Up	Shoulder In, Elbow Up, Wrist Down	Shoulder In, Elbow Down, Wrist Up	Shoulder In, Elbow Down, Wrist Down

**Table 10-4 – Arm reachability values**

Value		Description
Decimal	Binary	
0	00	IK and Collision issues
1	01	IK issues
2	10	Collision issues
3	11	No issues

## 10.3 Goodness Maps

Goodness maps analyze other products and compare their values to preset thresholds, reporting on how well the products meet their requirements. They generally use a scale of 0-5, where 0 indicates no data, 1-4 are various states of bad, and 5 is good (meets all thresholds).

Goodness maps should generally be used as a screening tool, to quickly look at the scene and determine what the candidates for surface interaction are. Users should look at the source data (reachability and roughness) if they want to determine how the answer was derived, or why a given location is not marked “good”.

For Mars 2020, there are three classes of goodness maps used for contact science. GSR is an intermediate map that contains roughness results; it generally is not used on its own.

GRN/GRA/GRC provide goodness values for each of the scenarios based only on arm reachability. GUN/GUA/GUC provide final goodness values for each scenario that combine all the elements – these are the ones users should concentrate on. See Figure 10-1. Note that the roughness products RUF/RUS/RUC also contain a status band that operates like a goodness map (see previous section). Table 10-5 defines the values used by the various goodness maps.

For GRN/GRA/GRC, the software looks at the relevant instruments for the scenario (see Table 10-2) and determines the worst result for those instruments for each configuration. It then looks at those results and returns the best result. In other words, a reachable pixel must be reachable by all relevant instruments in a single configuration. A second band indicates which configuration was best (most goodness maps are only one band).

The Helicopter also uses goodness maps but differently; see Section 14.2.2.3.

**Table 10-5 – Goodness Value Interpretation**

Value	GRN/GRA/GRC	GSR	GUN	GUA/GUC
0	IK+collision issues	Data issue	Data issue	Data issue
1	IK issue	Abrade and stab bad	Reach and rough issues	Reach and topo issues
2	n/a	Abrade patch issue	Reachability issue	Reachability issue
3	Collision issue	Stab ring issue	Roughness issue	Topography issue
4	n/a	n/a	n/a	n/a
5	Reachable, all tools	Abrade and stab OK	Reach and rough OK	Reach and topo OK

# 11. MOSAIC GENERATION

This section discusses the process of mosaicking multiple frames into a single RDR product using some projection. The text largely reflects the methods applied by MIPL. It should be noted that methods and software can differ between MIPL and other instrument teams; the algorithms followed by other teams may not be the same as described here.

## 11.1 Overview of Mosaics

Mosaics embody several important properties, making them very useful products. They assemble small pieces into a larger field of view, creating a better contextual view of the rover's surroundings. Certain projections create a level horizon, removing rover tilt. Mosaics can be calibrated so directions such as north and east can be determined - or they can be made relative to the rover to visualize forward and right. They can provide overhead views (Vertical, Polar, or Orthorectified projections) to help understand the local environment and place the instruments. They can be made from different types of data (such as slope). They can combine different filters to create color. They can coregister data from different cameras, including orbital images. Finally, they are the signature products for public outreach.

Mosaics can be assembled autonomously, or with manual assist to correct pointing errors and brightness mismatches. They can consist of two frames (or sometimes even one), or hundreds of images. In short, mosaics are among the most important imaging products for any *in-situ* mission.

## 11.2 Pointing Correction

An optional (but important) first step in mosaic production is pointing correction. This is used to minimize geometric seams (discontinuities) between frames. The results of pointing correction are used in mosaics, but they can also be fed back into the RDRs (often at the FDR stage), resulting in adjusted XYZ and other derived values and corrected meshes.

There are several methods by which improved pointing of the cameras can be determined. The most common method is to pick tiepoints between image pairs, either automatically or with manual assistance, and use those in a global cost function minimization to determine the corrected pointing parameters. Another possibility is to analyze the shape of XYZ data in the overlap region, again using an error minimization process to derive updated pointing parameters. Pointing parameters can also be determined manually.

Regardless of method, the result is encapsulated in a pointing correction or "nav" file. A more detailed description of this file is provided in Section 11.6, but fundamentally, this file contains, for each image being corrected, the original pointing parameters and the revised pointing parameters.

Pointing parameters are simply those numbers which represent how the camera is pointing in the rover frame, reduced to available degrees of freedom. These are used as inputs to the kinematics procedures which derive the camera model. The set of pointing parameters, together with the kinematics algorithm, is referred to as a pointing model. The job of a pointing model is to take a calibration camera model and transform it using the pointing parameters to create a transformed camera model which represents the specific image in question.

A given camera may have multiple available types of pointing models. The mast-mounted cameras have a standard model with two parameters: azimuth and elevation actuator angles. However, another model is available with three: azimuth, elevation, and “twist”, which is a rotation around the camera’s A axis (A being one of the CAHV camera model parameters). For arm-mounted cameras, there are two models: one with six parameters (the XYZ position plus the three Euler angles describing the orientation) and one with seven (XYZ plus the four components of a quaternion). The Hazcams have zero parameters, since they are rigidly attached to the rover body. Other pointing models may exist as well.

For the mosaic process, the “nav” file is sufficient to describe the pointing parameters. However, if other corrected RDRs need to be produced, such as XYZs or meshes, then the pointing parameters must be stored in the label. This is accomplished via the POINTING\_MODEL\_NAME and POINTING\_MODEL\_PARAMS labels. When recomputing a camera model, if these labels are present they should be used in preference to the normal method of pointing via labels in the ARTICULATION\_DEVICE\_STATE groups. For most users, however, the GEOMETRIC\_CAMERA\_MODEL should be used directly; this will be updated properly with respect to the corrected pointing.

### 11.3 Brightness Correction

Mosaics often exhibit visible differences in brightness and contrast at the seams between frames. This is reduced by the use of radiometric and color correction, which removes the effects of the camera and provides (in the case of radiometric correction) a measure of the actual light entering the camera. However, these do not take into account photometric effects – differences in scene brightness due to time of day or light scattering properties of the objects in the image. Even with perfect radiometric correction, there are often still visible seams due to photometric effects. Also, radiometric correction is not perfect; most notably, lens flares from looking toward the sun are not removed. For these reasons, many mosaics go through a brightness correction process.

The brightness correction process used by MIPL is similar in some respects to the pointing correction process. First, “tiepoints” analogues are gathered, called overlaps. In areas where two or more frames of the mosaic overlap, the mean and standard deviation of the overlapping regions are determined for each image. The size of the overlap area is limited, so if there is a large overlap between two images it will be broken into several smaller overlaps (this provides more opportunity for outlier rejection).

Then, an overall multiplicative (contrast) and additive (brightness) factor is determined for each image. The factors chosen minimize the “tiepoint error”, i.e. they attempt to match the statistics in the overlaps across all images. This uses a function minimization process similar to bundle adjustment. The number of pixels in each overlap is used to weight the overlaps such that each pixel has the same effect on the result. During this process, outlier overlaps are rejected. These outliers are common when the lander is being imaged, because of both the prevalence of specular reflections, and parallax effects.

Each image has a single multiplicative and additive factor applied to the entire image – corrections within the image are not performed in this algorithm. It can thus handle differences in intrinsic brightness fairly well, but does not work well with nonlinear effects such as uncorrected vignetting, photometric scattering differences, lens flares, etc.

The results are stored in a brightness correction file, which is analogous to the “nav” file in pointing correction.

For color images, this process is performed on the intensity channel after converting the image to HSI (hue, saturation, intensity) space. Hue and saturation are generally not corrected, nor do they generally need to be. It is possible to correct the R,G,B channels separately instead, but this has not worked well in practice.

It is also possible to correct using the mean and standard deviation of the entire image rather than overlap areas, but this is rarely done in practice.

Brightness correction is reduced in importance in M2020 compared to other missions by the development of Zenith scaling (Section 7.1.6), which serves to match brightness due to variations in solar illumination, thus taking care of the majority of brightness differences. However, it may still be used in some products.

## 11.4 Mosaicking Method for Most Mosaics

The process used by the MIPL software to create mosaics is described below. It consists of several sub-steps. Conceptually, one can think of the process as adjusting the pointing of the inputs, projecting them down to a surface, and looking at the result from a different point of view (the output projection). In reality, the process is run in reverse for ease of interpolation (this is described below).

### 11.4.1 Output Projection Determination

First, the output projection is determined based on the geometry of the inputs. The labels describing the projection are described in detail in Appendix B. The output projection parameters are determined by analysis of the inputs to give the “best” resulting mosaic, but can be overridden by the user. The determination process is outside the scope of this document; the results are what is important and they are in the label.

### 11.4.2 Surface Determination

A surface model is critical for mosaics (except for orthorectified mosaics, which don’t use one). This is a mathematical surface, which approximates the actual scene. To the extent that the scene differs from the surface model, distortion and uncorrectable seams due to parallax can result.

Usually the surface model is a flat plane, with normal pointing upwards. This can be adjusted, however, to better match the scene. Regardless, the results are documented in the SURFACE\_MODEL\_PARMS group.

There are five potential surface models in the MIPL software: PLANE, INFINITY, SPHERE, SPHERE1, and SPHERE2. See SURFACE\_MODEL\_TYPE in Appendix B for description. Note that an appropriate surface model is often determined automatically as part of the pointing correction process, and the surface model can be stored in the “nav” file. Almost all mosaics are created using the PLANE model.

### 11.4.3 Computation of Output View Ray

For each pixel in the output mosaic, a view ray in 3-D space is constructed. How this view ray is constructed depends on the projection type. In this section, the pixel is at location (i,j) in 0-based

coordinates, with *i* corresponding to sample and *j* to line. (0,0) is in the upper-left-hand corner. Capitalized values represent PDS label items from the SURFACE\_PROJECTION\_PARMS group. Unit and coordinate system conversions are applied as necessary but are not specified here. The coordinate system used is defined by REFERENCE\_COORD\_SYSTEM\_\* in SURFACE\_PROJECTION\_PARMS.

#### 11.4.4 Projection from Output to Surface

Once the view ray is determined, it is projected out until it intersects with the surface model. The resulting point in XYZ space is used in the next step. If the ray does not intersect the surface, the point is assumed to be at infinity in the direction the view ray is pointing. Exception: as mentioned below, the Vertical projection will reverse the direction of its view ray; infinity is assumed only if they both miss.

Note that the INFINITY surface model guarantees the ray will miss the surface at all times.

The difference between the SPHERE1 and SPHERE2 models is that, if the ray intersects the spherical surface more than once, SPHERE1 will take the first intersection, while SPHERE2 will take the second. For normal rover situations, SPHERE1 thus roughly models a convex hill, while SPHERE2 roughly models a concave crater when the rover is outside the sphere.

#### 11.4.5 Projection from Surface to Input

The XYZ location (or direction for the infinity case) is then back-projected into each input camera model in turn, using the corresponding input camera model. The first input for which the resulting pixel coordinate is inside the image (excluding border pixels which are thrown away) and non-0 stops the process; that is the image from which the output pixel value is taken. Values of 0 in the input image are ignored, with the effect that they are transparent.

This process results in stacking the images such that the first one in the input list of images "wins". There is no feathering of overlaps; the first image is "on top" of all the others, and an image completely covered by preceding images will not be used at all.

#### 11.4.6 Interpolation and Storage of the Result

Finally, a bilinear interpolation is (optionally) performed on the input image to reduce aliasing, based on the 4 pixels surrounding the back-projected location. The result of this interpolation is the value of the output pixel.

Bilinear interpolation is optional, but is normally done for image mosaics. Mosaics of other data types such as XYZ or Surface Normal (UVW) generally have interpolation turned off to avoid aliasing from interpolation with invalid pixels.

### 11.5 Mosaicking Method for Orthorectified Mosaics

The orthorectified mosaic is projected to a plane in a similar manner to the orthographic or vertical projections. However, unlike the other projections, the XYZ locations of the pixels are taken into account. This is what allows parallax to be removed. In a nutshell, the X and Y coordinates directly define where the pixel goes in the output, rather than projecting rays down to a surface model. The color, or DN, of the output comes from the associated "skin" image (or the Z value of the XYZ for

DEMs), which could be a traditional image or any RDR type. The challenges in this algorithm are dealing with overlaps, and appropriately sizing the pixels in the output to avoid salt-and-pepper effects (the farther away the pixels are, the farther apart they are in the orthorectified projection).

Unlike the other mosaic programs, ortho projection generation software program proceeds input-to-output. It loops through each pixel of each input. XYZ coordinates are converted to a common output coordinate frame as needed. The output pixel "hit" by this input pixel is determined. If two or more pixels (either from the same image or different inputs) "hit" the same output pixel, the "winner" is determined according to one of several algorithms:

- MIN, MAX: The pixel with the minimum or maximum Z value is chosen.
- AVG: The DN is computed as the average of all overlapping inputs.
- FIRST, LAST: The first or last image in the image list wins. FIRST mimics how all the other MIPL mosaic programs work.
- CLOSEST: The pixel closest to its camera is the one that is chosen.

Once the winners are determined, the regions around the pixels are filled in. The region potentially covered by an input pixel in the output is determined based on geometry, using the angular size of the pixel and distance to the camera. This determines how many output pixels can potentially be affected by this input pixel. The region size can be adjusted via parameters; generally it is enlarged in order to fill small holes due to missing pixels, while leaving larger holes in place. Any output pixel not directly "hit" is filled by looking at all such regions covering the pixel in question and using the one whose center is closest to it.

In reality, winner determination and filling are computed simultaneously, but the effect is as described above.

Afterwards, the program optionally performs morphological erosion to remove the repeated-rectangle pattern at the borders of the mosaic, which occurs when the pixel region is enlarged.

## 11.6 Mosaic Ancillary Files

A number of ancillary files are used to support mosaicking, and contain parameters and information describing how the mosaic was produced. With these, it is possible to maintain traceability and provenance for each pixel in a standard mosaic back to the source image. Ancillary files might not necessarily have the same basename as the mosaic, as several mosaics might share an ancillary file. In such cases, the ancillary file's name may indicate a different product type, projection, eye, filter/color, or geometric or brightness correction than the target mosaic's filename. The other filename fields should always match. The history label of the mosaic indicates which ancillary files were used.

It should be noted that many if not most mosaics are produced at least partially by hand, which explains most of the inconsistencies noted below. The general case is described, but as with any hand work not all conforms exactly.

These files are not described completely here, but we hope the descriptions are sufficient to be able to decipher them:

- a) **List files** - With a ".LIS" extension, list files are simple text files containing the names of the images making up the mosaic, one per line. The first name in the list references the image

This document has been reviewed and determined not to contain export controlled technical data.



frame that is “on top” in the mosaic product, covering the image frames that are referenced in the list below it. The list files often contain full pathnames to disks on the operational data store (ODS), directory paths which are not part of a PDS-released archive volume. However, the filenames themselves, minus directory paths, usually will be part of an archive volume. Occasional mosaics may have list files comprised of names for private copies of images (e.g. with “/home” in the pathname) that are the result of different scenarios of special processing; again, the file’s name and/or metadata label will identify the ultimate source image of the mosaic. Note that a mosaic’s input filenames (minus directory paths) is also contained in the mosaic’s label using keyword INPUT\_PRODUCT\_ID.

- b) **Nav files** - With a “.NAV” extension, nav files are XML files describing the pointing corrections that have been applied to images in a mosaic, as well as the surface model. The prologue contains identifying information.

Pointing correction works by applying a set of pointing parameters (e.g. mast azimuth and elevation, or XYZ position and Euler angles) to a kinematics algorithm and using that to re-point the camera model. See definition of keyword POINTING\_MODEL\_NAME in Appendix B for more information.

For each image (<solution> element), the Site and Drive components of the RMC are listed, followed by image identifier information. This identifier information contains the original pointing parameters, which allows the same correction to be applied to e.g. the left and right eyes (regardless of image ID information).

Following that are the updated pointing parameters, and then the revised (re-pointed) camera model. In rare cases, nav files are edited by hand, which puts the accuracy of the camera model update at risk. This is unlikely to occur for M2020, however.

At the end is usually a “surface\_model” element describing the surface model determined by the MIPL software program MARSNAV or MARSNAV2 (which create the nav files). This indicates the surface model used during nav file creation (MARSNAV2 does not use a surface model). This is then almost always the surface model used during mosaic processing (unless overridden on the command line). In any case, the SURFACE\_MODEL\_PARMS group of the mosaic product’s label describes the model used for the mosaic.

- c) **Tiepoint files** - With a “.TIE” or “.TPT” extension, tiepoint files are XML files containing image tiepoints used as input to program MARSNAV/MARSNAV2. These can be automatically or manually selected. The prologue relates each image ID to a key, which is used throughout the remainder of the file.

Each tiepoint has a left and right key, and then 1-based coordinates of the point in the corresponding files of the tiepoint. Just <left> and <right> should be used; <projected> has little value. In the flags, “quality” represents the quality of correlation match when the tiepoint was correlated (scale 0-1). The “interactive” flag has little archival value, it is used during the tiepoint gathering process. It does not (as the name suggests) indicate whether the tiepoint was automatically selected or manually tweaked.

Tiepoint type “0” is by far the most common and is a standard image tiepoint. Type 1 is occasionally used; it ties an image to a specific XYZ location (“fiducial” tiepoint); the coordinate frame for these points should be defined in the prologue. Other types are rarely used and their descriptions are beyond the scope of this document. Full descriptions are in the help documentation for VICAR program MARSNAV.

- d) **Brightness Correction files** - With a “.BRT” extension, brightness correction files are XML files containing information used to correct the brightness and contrast of images in a mosaic

This document has been reviewed and determined not to contain export controlled technical data.

relative to one another. They are similar in concept and structure to nav files. After the prologue, each image has one <brt\_solution>. The most common correction type, LINEAR, specifies an overall additive and multiplicative factor to be applied to each image (MULT is applied first, then ADD). These factors are echoed in the IMAGE\_RADIANCE\_FACTOR and IMAGE\_RADIANCE\_OFFSET keywords in the mosaic label. The HSI\_LIN type is similar, except the correction (for color images) is done in Hue-Saturation-Intensity (HSI) space, with the correction applied to Intensity only. See also BRIGHTNESS\_CORRECTION\_TYPE.

- e) **Brightness Overlap files** - With a “.OVR” extension, brightness overlap files are XML files containing information about image statistics in overlap areas, used to create the brightness correction files. They are similar in concept to tiepoint files, except the “tiepoints” are the mean and standard deviation of small areas of overlapping pixels for the mosaic.

They start with a prologue defining the image ID-to-key mapping, as with nav files. Each overlap then has a number of images involved in the overlap, the number of pixels, and a “radius” which is a general description of the maximum size of the overlap. This is followed by the key, mean and standard deviation of the overlapping area in each image. The line and sample coordinates are provided for an arbitrary point in the overlap, just to help locate where the overlap is. The actual shape of the overlap (or the location of the coordinate relative to the overlap) is not specified.

Overlap type “0” is a standard overlap. Type “1” gives the mean and standard deviation not of an overlap, but of the image as a whole (thus there is only one image underneath <overlap>). Type “2” is like type “0” but has mean and standard deviation in HSI space (intensity only). Type “3” is like type “1” but using HSI space. See the help documentation for MIPL software program MARSBRT for full details.

- f) **Rover filter files** - With a “.xmlf” extension, rover filter files are not associated with a mosaic; they go rather with the rest of the RDR files (particularly the “MXY” type). They are listed here for completeness, as they use the same ancillary label mechanism. See Section 8.11.

## 12. MOSAIC PROJECTIONS

### 12.1 Cylindrical Projection

Cylindrical projections are the most common method for viewing non-stereo panoramas.

The MIPL method for creating a Cylindrical projection involves computing the azimuth and elevation of the view ray, as follows:

$$\begin{aligned} azimuth &= i / MAP\_RESOLUTION + START\_AZIMUTH \\ elevation &= (ZERO\_ELEVATION\_LINE - j) / MAP\_RESOLUTION \end{aligned}$$

The view ray emanates from the point PROJECTION\_ORIGIN\_VECTOR.

Figure 12-1 shows an example cylindrical mosaic from MSL. Each pixel represents a fixed angle in azimuth and elevation. Rows are of constant elevation in the selected coordinate frame. In this case, a Site frame was used, so the horizon is level, and columns begin clockwise from Mars south (centered on north). Grid lines may optionally be added when the mosaic is created (most commonly, grids are “underneath” the image, showing only where the mosaic is blank).



**Figure 12-1: Cylindrical Projection**

### 12.2 Camera Point Perspective

Figure 12-2 shows a Camera Point Perspective mosaic. It is a perspective projection using a linear pinhole (CAHV) camera model. The mosaic behaves as though the "camera" which acquired the image frames was an instrument with a much larger field of view.

Point-perspective mosaics give the most natural view of small areas and are suitable for stereo viewing, but cannot be used for wide fields of view.

MIPL creates the Camera Point Perspective by using the output camera model (described by the GEOMETRIC\_CAMERA\_MODEL group in the output mosaic) to project the pixel into space. The origin of the view ray is thus the C point of the camera model, with the ray's direction being determined by the camera model.



**Figure 12-2: Point Perspective Projection**

## 12.3 Cylindrical-Perspective

Cylindrical-Perspective mosaics are used for large stereo panoramas, and work across a full 360 degrees of azimuth. Stereo is preserved because a baseline separation is maintained between the camera eyes at different azimuths.

This projection is the most complicated projection to create. Each column  $i$  (counting from 0) in the output mosaic is assigned its own camera model. This is done in several steps:

- 1) Compute initial camera model. This model is a CAHV linearized model derived from the first input to the mosaic, re-pointed to azimuth 0 and elevation PROJECTION\_ELEVATION. This model is stored in the GEOMETRIC\_CAMERA\_MODEL label group.
- 2) The instantaneous field of view of the "central" pixel (at the point where the  $A$  vector intersects the image plane) is computed using the formula:

$$ifov = atan(1.0 / |(\vec{H} - \vec{A} * (\vec{H} \cdot \vec{A}))|)$$

where the “•” indicates the scalar dot product of the two vectors  $A$  and  $H$ .

Alternatively, this can be derived from the image size and azimuthal extent (where the azimuths are adjusted by 360 degrees such that the result is

This document has been reviewed and determined not to contain export controlled technical data.

minimally positive):

$$ifov = (STOP\_AZIMUTH - START\_AZIMUTH) / LINE\_SAMPLES$$

- 3) The azimuth of the column is computed:

$$azimuth = START\_AZIMUTH + i * ifov$$

- 4) The initial camera model is re-pointed using kinematics as described above under the pointing correction section, using the above azimuth and PROJECTION\_ELEVATION. This results in the final camera model for the column.

Step 4 is difficult to duplicate for reconstructing the set of camera models. For that reason, an alternate method is described in this paragraph. The resulting models are exact for mast-mounted cameras with no backlash correction; they are a close approximation for other cases. In general, for mast-mounted cameras, the C points of the column camera models describe a ring in space, whose diameter is approximately the baseline between the cameras. This ring is described by PROJECTION\_ORIGIN\_VECTOR (center), PROJECTION\_AXIS\_OFFSET (radius), and CAMERA\_ROTATION\_AXIS\_VECTOR (orientation of the ring axis). These together simulate the kinematics motion of a mast-mounted camera in lander frame. To compute the camera model for the azimuth defined in Step 3, take the camera model from the label, and rotate the entire camera model around the camera rotation axis by the azimuth amount, using the ring center as the pivot point. The C point will remain on the ring, while the camera pointing (close to but not identical to the A vector) will remain approximately tangent to the ring at that point. After this, compute the rotation required to transform CAMERA\_ROTATION\_AXIS\_VECTOR into PROJECTION\_Z\_AXIS\_VECTOR (which can be done by taking the cross product to get the rotation axis and the dot product to get the rotation amount). Then rotate the camera model by this amount, again using the ring center as the pivot point. This has the effect of tilting the entire ring so it is perpendicular to the PROJECTION\_Z\_AXIS\_VECTOR. This last rotation is often used to remove the effect of lander tilt, resulting in a flat horizon with the camera model baselines (vector between the left and right eyes) aligned with the horizon (technically, perpendicular to the Z axis in Local Level frame). For this “untilt” case, the PROJECTION\_Z\_AXIS\_VECTOR is the Local Level frame’s Z axis expressed in Lander frame. Note that PROJECTION\_ELEVATION and PROJECTION\_LINE are measured before this “untilt” rotation takes place, so they end up describing a sinusoid in the final mosaic when untilt is used.

Once the camera models have been defined, the mosaic proceeds through each pixel as with the other projections. The view ray is computed as described below (A, H, and V come from the column's camera model):

$$\begin{aligned} x\_center &= \vec{A} \cdot \vec{H} \\ y\_center &= \vec{A} \cdot \vec{V} \\ samp &= x\_center \\ line &= y\_center + j - PROJECTION\_ELEVATION\_LINE \end{aligned}$$

where the “•” indicates the scalar dot product of two vectors. This (samp, line) coordinate is then projected into space using the column's camera model, and this projection becomes the view ray. The origin of the view ray is the column's C point.

Figure 12-3 shows a Cylindrical-Perspective projection in which a 360 degree view can be viewed in stereo. This is a perspective projection similar to Section 12.2 except that the mosaic acts like a pinhole camera which follows the mosaic in azimuth while maintaining camera baseline separation. If the mosaic is generated with no tilt correction (i.e., `CAMERA_ROTATION_AXIS_VECTOR` and `PROJECTION_Z_AXIS_VECTOR` are the same) and the rover is tilted, the horizon will not be level, instead being sinusoidal. This preserves epipolar alignment and allows for better stereo viewing of the panorama. However, for aesthetic reasons, Cylindrical-Perspective mosaics are often created by “untilting” the rover as described above. In these cases, the horizon will be level, but stereo alignment may be compromised due to parallax effects in areas where the surface model does not closely match the actual surface. Additionally, the overall baseline between the cameras may be adjusted via the ring radius (`PROJECTION_AXIS_OFFSET`). This has the effect of enhancing or reducing the overall disparity, which can result in better stereo viewing in some cases. This baseline adjustment may create similar parallax effects in areas where the surface model does not match the actual surface.



**Figure 12-3: Cylindrical-Perspective Projection**

## 12.4 Polar Projection

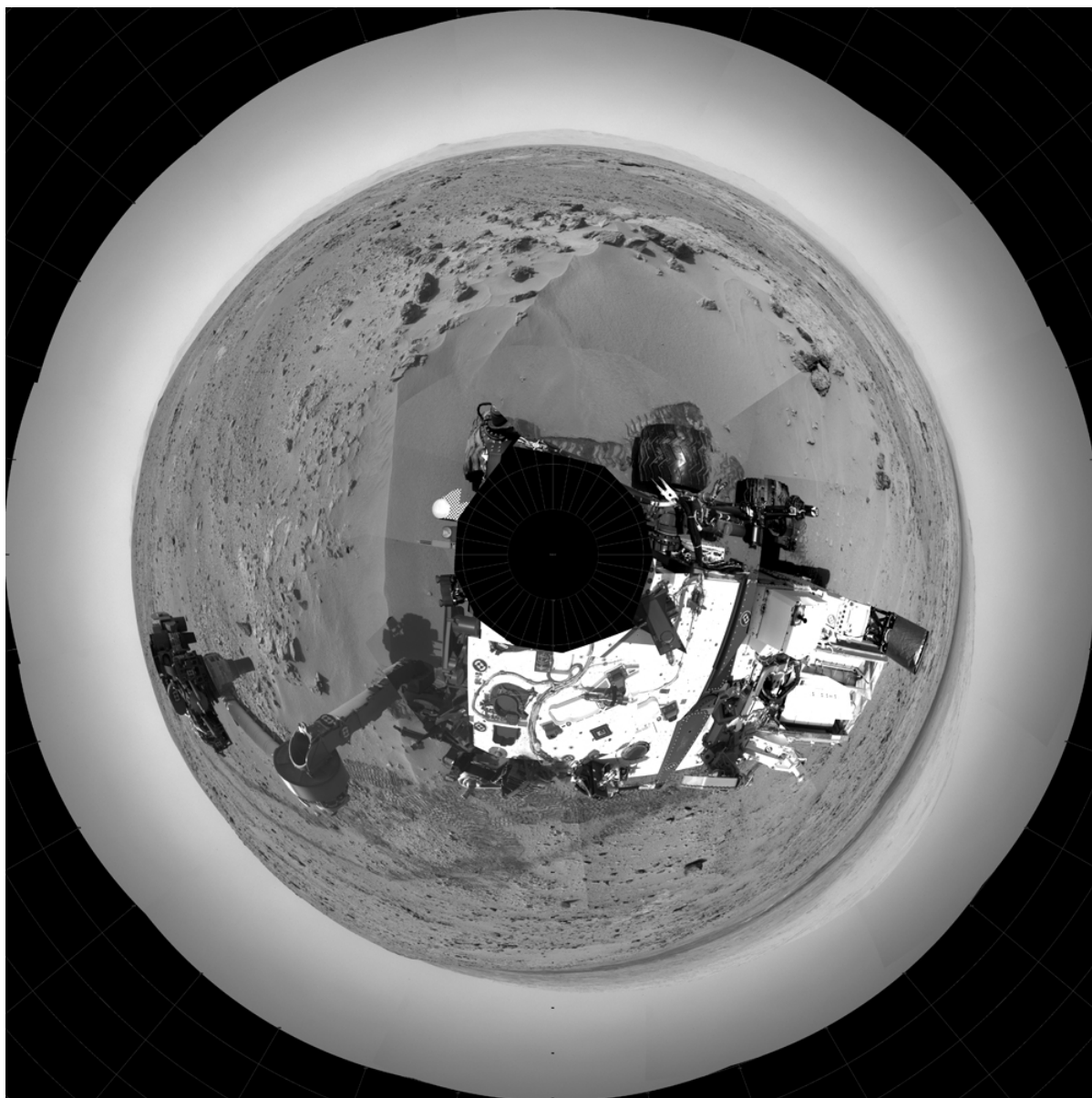
Polar mosaics create a quasi-overhead view that still allows viewing all the way to the horizon.

MIPL creates the Polar projection by computing the azimuth and elevation of the view ray as follows:

$$\begin{aligned}
 x &= i - \text{SAMPLE\_PROJECTION\_OFFSET} \\
 y &= \text{LINE\_PROJECTION\_OFFSET} - j \\
 \text{range} &= \sqrt{x^2 + y^2} \\
 \text{elevation} &= \text{range} / \text{MAP\_RESOLUTION} - 90 \text{ degrees} \\
 \text{azimuth} &= \text{REFERENCE\_AZIMUTH} + (90 \text{ degrees} - \text{atan2}(y, x))
 \end{aligned}$$

The view ray emanates from the point `PROJECTION_ORIGIN_VECTOR`.

Figure 12-4 shows a Polar projection. Concentric circles represent constant projected elevation. Mars nadir is at the convergent center and north is up.



**Figure 12-4: Polar projection**

## 12.5 Vertical Projection

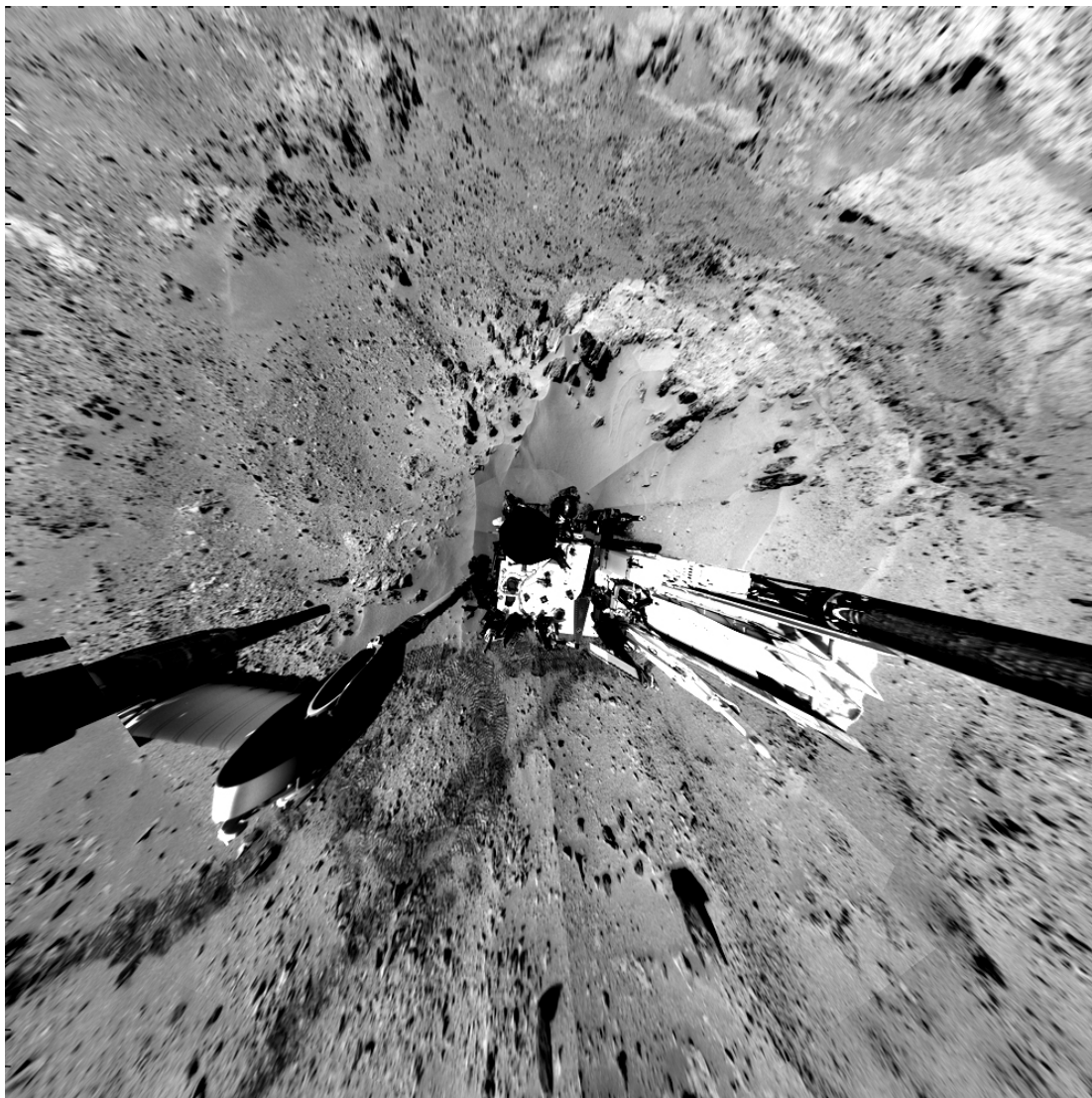
Vertical mosaics provide a view of the surroundings as if you were looking straight down. They are thus quite useful for establishing the environmental context or comparing with orbital images, but suffer from severe distortion with any variance of the scene from the surface model. In particular, rocks are severely elongated, and the terrain is not taken into account (contrast this with Orthorectified, Section 12.7, where terrain is taken into account).

MIPL creates the Vertical projection as follows:

$$\begin{aligned}nl &= \text{number of lines in the mosaic (LINES)} \\ns &= \text{number of samples in the mosaic (LINE_SAMPLES)} \\x &= (nl/2 - j) * MAP\_SCALE \\y &= (i - ns/2) * MAP\_SCALE\end{aligned}$$

The view ray emanates from  $(x, y, 0)$  and points straight down  $(0,0,1)$ . If the ray misses the surface, it is changed to point straight up  $(0,0,-1)$ .

Figure 12-5 shows a vertical view. It assumes that the field is a plane tangent to the Martian surface with up pointing north. This is not an orthorectified rendering but is still useful in many situations.



**Figure 12-5: Vertical Projection**

## 12.6 Orthographic Projection

The Orthographic projection is a generalization of the Vertical projection. It differs from Vertical in that an arbitrary projection plane can be specified.



If  $O$  is the point specified by the `PROJECTION_ORIGIN_VECTOR` and  $\hat{X}$  and  $\hat{Y}$  are the unit vectors given by `PROJECTION_X_AXIS_VECTOR` and `PROJECTION_Y_AXIS_VECTOR` respectively, then an arbitrary point  $P$  will have projection coordinates  $(X, Y)$  as follows:

$$\begin{aligned} X &= (\vec{P} - \vec{O}) \cdot \hat{X} \\ Y &= (\vec{P} - \vec{O}) \cdot \hat{Y} \end{aligned}$$

where the “ $\cdot$ ” indicates the scalar dot product of two vectors. `PROJECTION_Z_AXIS_VECTOR` is the direction of projection; the three vectors form a right-handed orthonormal basis.

All of these quantities must be specified with respect to a single frame defined by the `REFERENCE_COORD_SYSTEM_NAME` and `REFERENCE_COORD_SYSTEM_INDEX`. Additional relevant parameters for the projection are `MAP_SCALE`, `X_AXIS_MINIMUM`, `X_AXIS_MAXIMUM`, `Y_AXIS_MINIMUM`, and `Y_AXIS_MAXIMUM`.

A Vertical projection is the same as Orthographic with `PROJECTION_X_AXIS_VECTOR` = (1,0,0), `PROJECTION_Y_AXIS_VECTOR` = (0,1,0), and `PROJECTION_Z_AXIS_VECTOR` = (0,0,1).

## 12.7 Orthorectified Projection

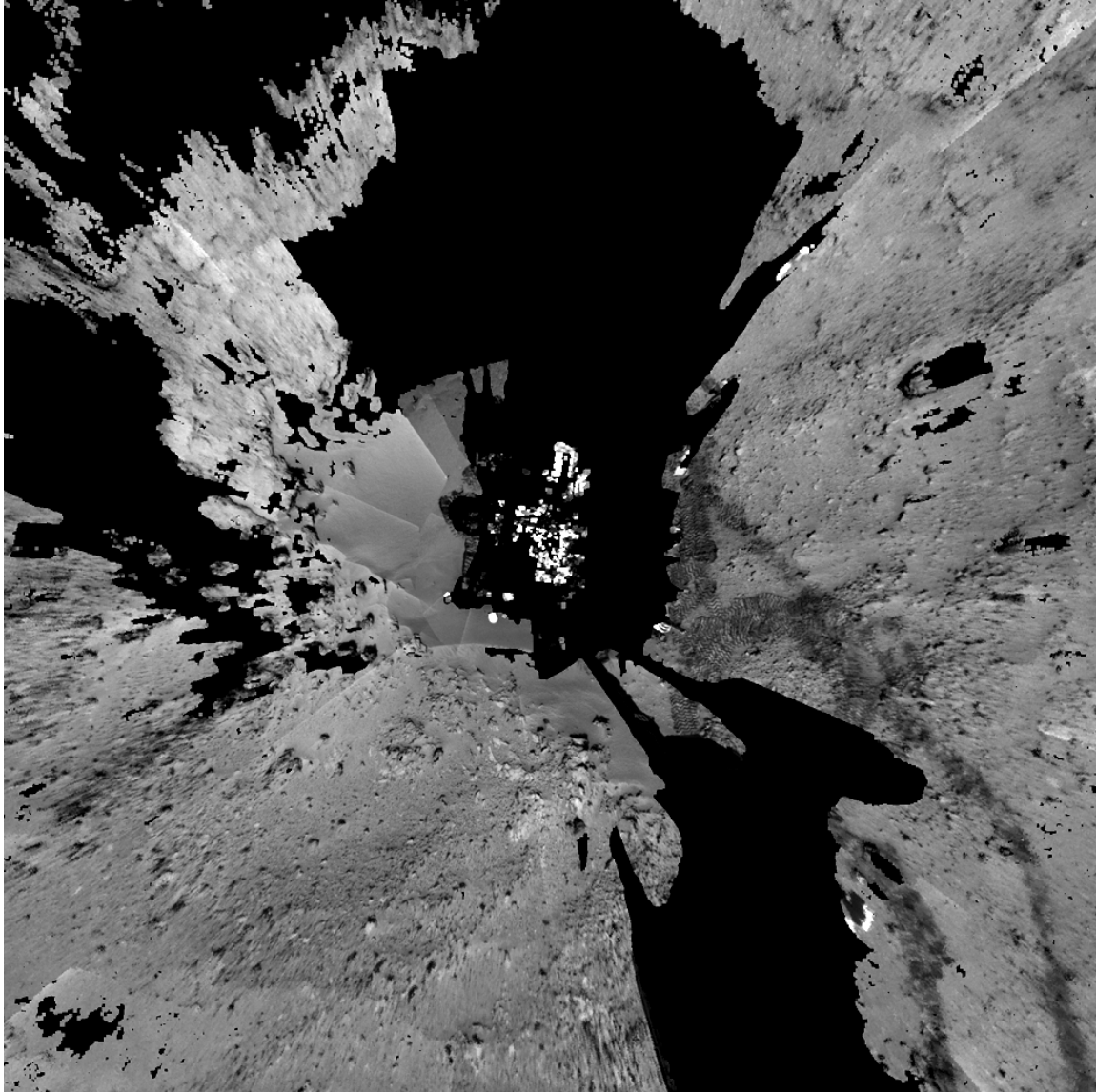
Orthorectified mosaics are used to show a “true” view of the scene from a different point of view, without distortion due to parallax. The point of view is usually overhead, resulting in an image suitable for comparison to satellite images and mapping. The removal of parallax necessarily leads to holes or gaps in the mosaic, which do not occur with the other projections. This projection is often used for rover localization, to compare against the orbital base map (Section 15.2).

The Orthorectified mosaic is projected to a plane in a similar manner to the Orthographic or Vertical projections. However, unlike any of the other projections, the XYZ location of the pixels are taken into account. This is what allows parallax to be removed. See Section 11.5 for algorithm details.

A simple way to think of this, for the case of an Orthorectified-Vertical projection, is to attach the XYZ coordinate (derived from stereo analysis) to each input image pixel, chop off the Z coordinate, and use the XY coordinates as the position in the output image. The more general (non-vertical) case is similar in concept, just rotate the XYZ values to the frame defined by the projection plane first.

The most common projection plane is to look straight down, which corresponds to the same point of view as the Vertical projection. For this case, `PROJECTION_X_AXIS_VECTOR` = (1,0,0), `PROJECTION_Y_AXIS_VECTOR` = (0,1,0), and `PROJECTION_Z_AXIS_VECTOR` = (0,0,1).

See Figure 12-6 for an example orthorectified mosaic.



**Figure 12-6: Orthorectified Projection**

## 12.8 Non-image Mosaics

Normally mosaics are created using traditional images, where each pixel is either a raw or radiometrically corrected intensity value. However, mosaics can also be created using other types of pixels. In fact, any of the RDR types using an image format (e.g. not meshes) can be mosaicked. Among the most useful of these are mosaics of XYZs, which form the basis for the Helicopter placement products (Section 14.2.2).

For example, an XYZ mosaic contains XYZ values for each pixel in the mosaic rather than intensity values. The inputs to the mosaic program are XYZ files (or individual X, Y, or Z components), and the pixels are interpreted in the same way - as the coordinate of the corresponding pixel in Cartesian space. Like XYZ images, the mosaics may consist of a single 3-band file with X, Y, and Z components, or separate 1 band files for each component. A Z-only mosaic of a Vertical or Orthorectified-Vertical projection creates a digital elevation model (DEM) - approximate in the Vertical case, correct for Orthorectified-Vertical. Note: such DEMs can be created with the standard +Z down, in which case they use the “ZZZ” type, or with +Z up, in which case they use the “DEM” type code.

Care must be taken while producing these mosaics to ensure that a consistent coordinate system and data type are used for all the input images. No transform is done on the data; the output mosaic may have only one coordinate system in which the values are defined, and one DERIVED\_IMAGE\_TYPE.

Non-image mosaics are often created without interpolation; the nearest pixel is used instead. This avoids aliasing effects when pixels are interpolated with neighboring invalid pixels.

## 12.9 Mosaic Index Files (ICM/IDX)

On occasion, mosaic index files (IDX, ICM) will be created to go along with mosaics. These files are the same size and geometry as the corresponding mosaic. Together, they indicate where each pixel came from in the source images.

IDX (index) files show which image the pixel came from. The value is an integer, 1-n, which corresponds to the position of the image in the input .LIS file (and also the position in the INPUT\_PRODUCT\_ID label). If two images overlap, the one “on top” (the mosaic programs do not generally blend images) is the one shown in the index file.

ICM (coregistration map) files show which pixel in that image the input came from. Like a disparity map, the two bands indicate the (1-based) line and sample (respectively) coordinates of the input pixel. The coordinates are floating-point values, because the output pixels rarely map to integer input coordinates. The pixel value actually used depends on the mosaic algorithm, but is generally bilinear interpolation.

## 12.10 Pixel Scale for Mosaics

Section 8.15 discusses how to determine pixel scale for images. For mosaics, this varies based on the type of projection.

For Perspective mosaics, the same process is used as for images, although the ifov must be computed from the camera model, as it can vary. Perspective mosaics are often generated at the same scale as their first input (in which case the Section 2 tables could apply), but this is not guaranteed.

Vertical and Orthorectified mosaics are easy: the pixel scale is constant across the image, and is defined in the MAP\_SCALE label (meters/pixel).

For cylindrical mosaics, the ifov can be obtained from the MAP\_RESOLUTION label. This is given in pixels/degree, so convert it to radians/pixel. The range can be determined using any of the same methods as for single images. If the assumed surface model method must be used, note that the surface model used for the mosaic itself is defined in the label.

Cylindrical-perspective mosaics can be treated the same as cylindrical. The MAP\_RESOLUTION label applies only in the horizontal direction, but that should be sufficient for most purposes. The vertical direction, if needed, can be derived from the camera model.

## 13. OTHER PROCESSING

### 13.1 Terrain Classification

Terrain Classifier RDR products attempt to identify different types of terrain using a neural network analysis of image data. They are used operationally to improve predictions of rover slip for planning drives by attributing a different set of slip parameters to each terrain type.

The TEN Terrain Probability map is a multiband product with one band per terrain class, where the DN value of a given band corresponds to the confidence score returned by the classification algorithm. It is registered geometrically to the source FDR. The total number of bands is not fixed in order to allow flexibility for new terrain classes to be added. Classes are defined by the CLASS\_DESCRIPTION VICAR/ODL keyword, and may represent geological features such as sand, small rocks, bedrock, etc., or rover traversability characteristics like high slip, medium slip, etc. There is a maximum range parameter, which describes the maximum distance at which terrain classes were applied, as the classification performance varies with distance from the camera. The maximum range is applied on a line-by-line basis.

The Terrain Classification map is a 2-band product that is derived from the TEN map, where the first band is a map of the most probable terrain class for each pixel, and the second band is a confidence score.

### 13.2 Incidence, Emission, Phase

The IDP type contains incidence, emission, and phase angles for each pixel for use in photometry work. It is a 3-band float product derived from UVW in the order (incidence, emission, phase). Incidence angle is the angle between the sun and the surface normal. Emission angle is the angle between the surface normal and the camera view (vector between the camera C and the pixel). Phase angle is the angle between the sun and the camera view vector.

### 13.3 Image Overlays

Many image RDR types represent some quantity other than intensity of light, such as XYZ or slope. The value at each pixel indicates the measurement of the quantity at the corresponding point in the original image. These types can be overlaid on a background of the EDR or other intensity image, using color coding to represent the RDR value. This gives a visualization of the RDR in context of the scene. Overlays are generally indicated by a product type ending with “O” in the filename.

### 13.4 Filled Images

Many RDR types do not achieve full coverage, e.g. the correlator is unable to find a solution at every point or the XYZ point failed the various filters. These “holes” are preserved in the RDRs using some value to indicate no solution (see the MISSING\_CONSTANT labels). For ops work, it is critical to know where the holes are, so they are preserved in all nominal RDRs. Some RDRs created for purposes other than operations have these holes filled in using an interpolation mechanism. These are referred to as “filled” RDRs. Filled RDRs are generally indicated by a product type ending with “F” in the filename (but not all trailing “F”s mean Filled).

## 14. INSTRUMENT-SPECIFIC PROCESSING

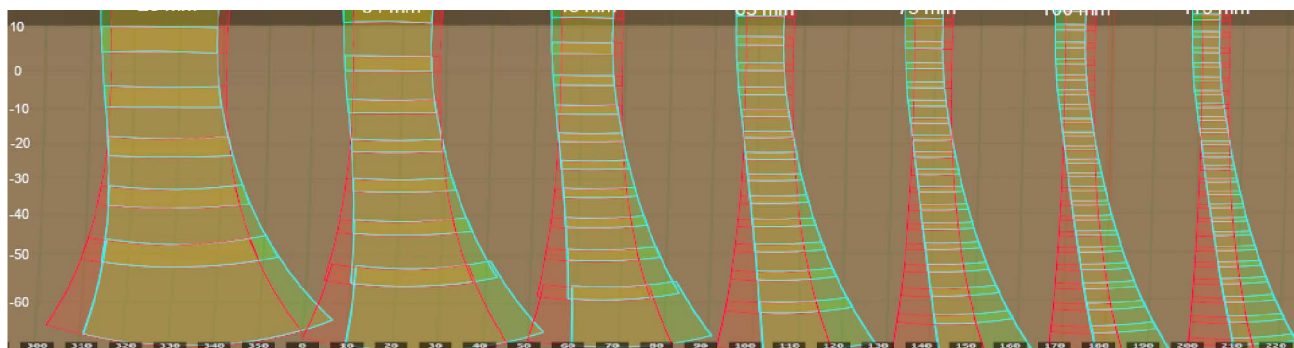
### Mastcam-Z

The Mastcam-Z is a stereo camera system with a variable zoom but a fixed toe-in of 1.15 degrees per eye. This fixed toe-in means that the stereo overlap is variable both as a function of range and of zoom setting. Figure 14-1 illustrates this. The orange areas indicate stereo overlap at each zoom setting as a function of range (using elevation as a proxy, assuming a flat plane surface). The red and green fringing indicate areas of the image that do not have stereo overlap.

Although not a problem at lower zooms, the stereo overlap becomes highly problematic at higher zooms and close-in targets. For this reason, many Mastcam-Z observations (at high zooms of close targets) will use some form of re-pointed stereo rather than nominal simultaneous stereo. It should be noted that the Navcam has a wide enough field of view and low enough zoom that this is not an issue for Navcam, so repointed stereo will be much more rare.

Repointed stereo means repositioning the mast to take the stereo partner. This leads to greatly increased stereo coverage, but at the expense of accuracy. Stereo range accuracy is largely determined by knowledge of the camera baseline (distance between the cameras). With single-pointing stereo, the baseline is fixed by the mechanics of the camera head. While the pointing may be somewhat inaccurate, the baseline is fixed and highly accurate. Repointed stereo however means the mast moved in between exposures – thus the baseline is variable and knowledge of it depends on knowledge of the actuator positions or by ground-based bundle adjustment. Both of these are less accurate. Users should be aware of this distinction when using stereo data.

Despite the less accurate results, repointed stereo is still highly useful and is accurate enough for many uses.



**Figure 14-1: Mastcam-Z Stereo Coverage**

mosaic acquisition.

- 1) Standard stereo overlap. Single-pointing stereo is used throughout, with pointings adjusted to achieve full stereo overlap. This is the nominal mode for Navcam, and the likely mode for low-zoom Mastcam-Z.

This document has been reviewed and determined not to contain export controlled technical data.

- 2) Repointed stereo pairs. In this mode, each frame is taken as a repointed stereo pair independently, with frames close enough together to achieve full stereo overlap. Stereo comes only from the matched pairs.
- 3) Independent mosaics. In this mode, independent mosaics are taken from the two cameras. Each eye achieves full coverage itself, without regard to stereo. Stereo analysis requires pairing each image with more than one partner (typically, two, as the “tiers” of a mosaic generally line up even in this mode).

The mechanism described in Section 8.1 is used for cases 2 and 3 to indicate the stereo partners.

## 14.2 Helicopter

There are two kinds of processing for the Helicopter. The first is for images from the helicopter itself. The second is for images from the rover that are used to support helicopter placement. Both are described here.

### 14.2.1 Helicopter Images

The helicopter has two cameras: a grayscale Navigation camera and a large color “Return To Earth” (RTE) camera (Section 2.9).

The helicopter is a technology demonstration. As such, there are no specific requirements on processing the images, other than creating EDRs for both cameras. However, a variety of products were made during operations, both as a test of IDS capabilities and as a demonstration of what can be done with helicopter data.

The specific RDR products produced from the helicopter data cannot be predicted in advance. However, they will likely include stereo analysis products (both from lateral and vertical motion), meshes, landing site localization, and mosaics. All such products will conform to the conventions in this SIS.

Note that the RTE camera sometimes does white balancing on board and sometimes doesn’t, with no metadata indicating the difference. The pipeline attempts to detect this using image analysis, and applies a different set of radiometric correction parameters to make the images appear similar. See Appendix C.

### 14.2.2 Helicopter Placement Support

In order to safely fly the helicopter, an “airfield” and flight zone must be found and characterized. The airfield defines the landing areas, while the flight zone supports overflights. These zones must meet certain criteria in order to be usable for helicopter operations.

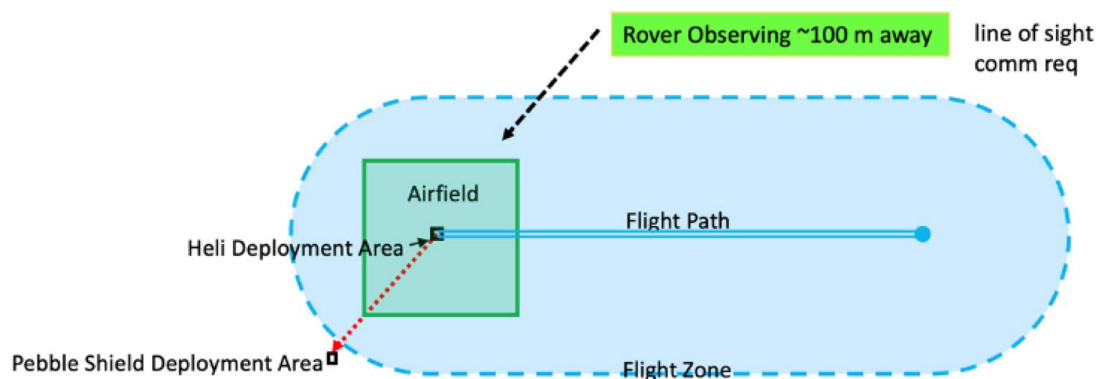
Orbital data will be used to help pre-screen potential sites. However, orbital data is not high enough resolution to certify the deployment and flight areas. Therefore, rover imagery will be required.






There are four areas of interest for helicopter operations:

- Flight Zone: the area in which helicopter flights will take place. Must meet requirements related to slopes, rocks, and visual feature richness (to support onboard visual navigation software).
- Airfield: A 10x10m area at one end of the flight zone, centered on the helicopter dropoff location. Serves as the nominal takeoff and landing area and thus has stricter slope/rock requirements.

- Helicopter Deployment Area: A 3x3m area at the center of the airfield where the helicopter is deployed. Has even stricter requirements.
- Pebble Shield Deployment Area: A 3x3m area outside the flight zone where the pebble shield is dropped to the ground.

These zones are shown in Figure 14-2 along with the high level requirements for rocks and slopes.



Symbol	Region	Size/Location	Slope	Rock Tolerance
	Pebble Shield Deployment Area	≥15m from Heli Dply., 3m x 3m area	<10° over rover lengthscale	5 cm Maximum Ground Clearance Reduction
	Heli Deployment Area	3m x 3m area	<6° over rover and heli lengthscale	No rocks/relief ≥5 cm, Max Ground Clearance Reduction
	Airfield	10m x 10m area	<6° over heli lengthscale	Req: <13 rocks ≥5cm high
	Flight Path	15m on each side of flight path	<6° over 2m	Desire: <13 rocks/relief ≥5cm high per 100m <sup>2</sup>
	Flight Zone		<4° over 10m	
			<2° over 20m	
			<1.5° over 40m	
			<1° over 80m	

v.5

In order to analyze these requirements, orthorectified mosaics are made from rover data (primarily Navcam) and then analyzed. Note that the flight zone slope  $\geq 10m$  requirements are analyzed using orbital data and are not discussed further here.

These placement products will be made systematically at every post-drive location until the helicopter operations are complete. This allows the team to understand better the placement products during the early mission, even before getting to potential flight zones.

### Helicopter Orthorectified Mosaics

Orthorectified mosaics are created at two scales:



- 30 m radius, 1 cm/pixel, “30M01CM”
- 5 m radius, 2.5 mm/pixel, “5M2P5MM”

The quoted name is what is included in the description field of the mosaic filename.

For each scale, three actual mosaic products are created:

- RAS (or other image type): the orthorectified image itself
- ZZZ: the DEM, single-band elevation map with +Z down
- XYM: the full masked XYZ coordinate for each pixel in the mosaic, used for analysis products

Additionally, a slope map (SLP) is made at a 2m diameter from the XYM mosaic.

Note that these orthorectified mosaics are usable generally, not just in the context of helicopter placement products.

#### 14.2.2.2 Helicopter Placement Products

There are two placement products that are derived from InSight software [NSYT\_Cam\_SIS]. These products model how the helicopter legs interact with the terrain.

The TFH product (Tilt For Heli) is a measurement of the min and max tilt if the helicopter landed at that pixel. It considers all clock angles (rotations around the vertical axis) since that may not be well controlled. At each clock angle, it looks at where the feet sit on the ground, and calculates the tilt of the helicopter given that. So if a leg would be perched on a rock, that gets a high tilt. This is distinct from slope – slope is a measure of the ground, while tilt is a measure of how the feet interact with the ground.

The RUH product (Roughness For Heli) looks at surface roughness under the body as a whole, between the legs. Thus, if the helicopter straddles a rock, it gets a high roughness value. It looks only at positive excursions (rocks) – depressions are ignored. If the helicopter perfectly straddles a rock, it could have a high roughness but zero tilt, which is why both products are needed.

Both files are 3-band float files, where the first band is a status or goodness value (0-5 indicating whether it met thresholds) and the other two bands are the actual values. For RUH the third band is always 0 (it is intended for measuring roughness under the feet, but that is not used for the helicopter). See Table 14-1 for the band assignments for these products, and Table 14-2 for the meaning of the goodness values.

**Table 14-1 – Helicopter Placement Product Bands**

Band	TFH meaning	RUH meaning
1	Goodness	Goodness
2	Minimum Tilt	Body Roughness
3	Maximum Tilt	All 0

**Table 14-2 – Helicopter Goodness Values**

Value	TFH meaning	RUH meaning	SLH meaning	GUH meaning
0	No data	No data	No data	No data
1	n/a	n/a	Slope $\geq 7$ deg	Multiple bad
2	n/a	Roughness bad	n/a	TFH bad
3	Tilt bad	n/a	$6 \leq \text{slope} < 7$ deg	RUH bad
4	n/a	n/a	n/a	SLH bad
5	Tilt good	Roughness Good	Slope $< 6$ deg	All good

#### 14.2.2.3 Helicopter Goodness Products

The helicopter goodness products are similar to the arm goodness products discussed in Section 10.3. They contain a value from 0-5 that indicate whether the given point meets requirements or not, with 5 indicating all requirements are met, and other values indicating problems. See Table 14-2 for values.

Goodness maps should generally be used as a screening tool, to quickly look at the scene and determine what the candidates for helicopter placement are. Users should look at the source data (tilt, roughness, and slope) if they want to determine how the answer was derived, or why a given location is not marked “good”.

The SLH product is a goodness map for slope. It simply shows whether the area meets the 6 degree slope requirement, is close (7 degrees), or not. Slope is distinct from instrument tilt in that slope is a characteristic of the terrain itself, while instrument tilt describes how the feet interact with the terrain.

The GUH product is the primary goodness map for the helicopter. It combines the TFH, RUH, and SLH products together into a single goodness value for this location (at the given size scale; there's a GUH for each of the 30m and 5m mosaics).

#### 14.2.2.4 Extended Helicopter Operations

The above products were used extensively during the 5-flight helicopter demonstration period during operations. After that, the helicopter continued to fly, transforming from a technology demonstration into a scouting mission for the rover and science teams. During this extended mission, the rover did not pre-survey the airfields in order to characterize them. Rather, orbital images were used to determine safe places to land.

During this extended mission, the helicopter routinely acquired a navcam “carpet”, consisting of full coverage with overlaps looking straight down along the flight path. These were processed into orthomosaics and DEMS along the flight path. The last few images of each flight were vertically projected in order to aid in localizing the landing spot (by comparing features to an orbital map). Additionally, RTE images were acquired along the way during flight. These were also vertically projected to create color maps of a wider area around the helicopter. On occasion, RTE images were taken with a geometry that was conducive to stereo analysis; for these a complete set of stereo processing, all the way to meshes, was performed.

These special products are all included with the helicopter delivery to PDS. As of this writing, georeferencing information is not included, but that may change in the future. They are all located

with respect to a rover frame (typically Site 3), but there are errors in this location that may or may not be accounted for in the products themselves.

Other types of products may be created for the helicopter as operations proceed.

## 14.3 PIXL MCC Trapezoid Correction

The PIXL MCC camera is mounted at approximately an 18° angle with respect to the boresight of the X-ray source. Since the X-ray is almost always pointed along the local surface normal, this introduces a distortion into the MCC images. This results in rectangles appearing as trapezoids.

It should be noted that the camera model correctly describes the image as-is. Thus the original geometry can be used in mosaics, coregistration, and any other activity. However, for geologic interpretation of the scene, the trapezoidal distortion is undesirable. For this reason, products are created that remove this distortion.

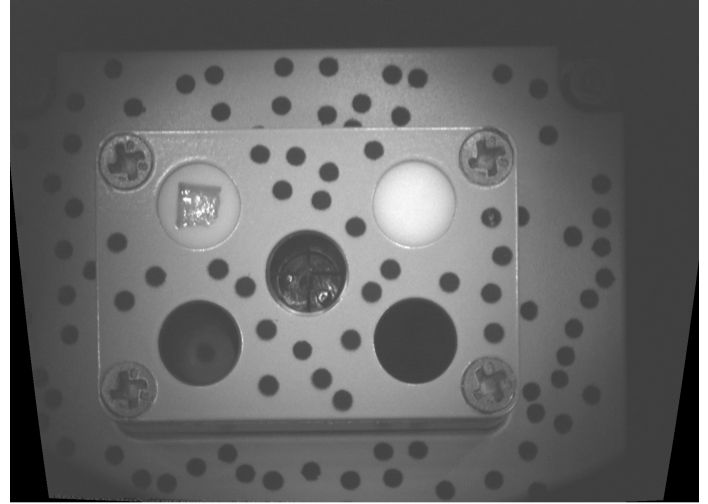
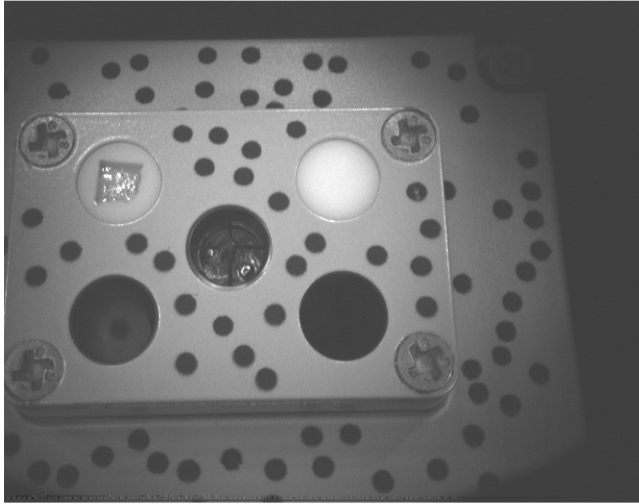
Figure 14-3 shows an example. The image on the left is a raw image of a rectangular calibration target. The image on the right is the result of trapezoidal correction, using a plan surface model. Note that the target is again rectangular.

Such trapezoid-corrected products are indicated by a “T” in the filename geometry field. Any product type (3-letter type code) may have trapezoidal correction applied.

Trapezoidal correction works by projecting the image to a surface and back into a different virtual camera (the same process as is used for mosaic projection, see Section 11.4). The virtual camera is aligned with the X-ray source, looking square on to the surface.

There are two options for determining the surface model: plane and mesh. Both are derived from the SLI (structured light) mechanism built in to PIXL. For the plane surface, the SLI results are fit to a plane that best describes the surface. For the mesh method, the SLI results are used to create a small terrain mesh, which is used as the projection surface. The mesh method is likely more accurate, as the mesh follows the actual contours of the terrain. However, the plane method is likely more robust, as small errors in the mesh won't cause errors in the projection.

Either way, the projection suffers from parallax distortion, the same as any other mosaic. The closer the surface is to the surface model, the less distortion there will be.



## MEDA SkyCam

SkyCam acquires images differently from most cameras. Rather than being under direct sequence control, in baseline operations the images are acquired in the background by the instrument, and then retrieved by a sequence at a later time, along with other data. Since different sequences can be used for this purpose, there is no direct relationship between the sequence ID and the image. Therefore, for the PDS archive and for later operations, the sequence ID is hardcoded to “MEDA00000”.

For similar reasons, the SkyCam label is extremely sparse. It was originally populated with information from the downlink, but this turned out to be confusing as the downlink was sometimes days later than the image acquisition. Therefore, all of this misleading metadata is now suppressed from the label. The minimal label includes the information that is reliable from telemetry. It also includes information gleaned from the PLACES database (Section 15.3), which is used to translate the SCLK to the proper site and drive, and then to extract the rover pose.

## Cruise Flight Software Idiosyncracies

The first few sols of the mission (0-4) were run on the “Cruise FSW”, which had very limited surface capability due to the needs of Entry, Descent, and Landing. On sol 9 the FSW was transitioned to the Surface version. Those first few sols thus have a number of differences with respect to the rest of the mission.

- The sequence ID is always AUT\_04096
- Automatic pipeline reconstruction of ECAM tiles was not performed. Manual reconstruction (with Special Processing Flag set) was done.
- SuperCam images do not have FITS files (there are 4 of them)

In addition, several metadata (label) items are incorrect or unreliable. A partial list is included below, see the PDS Release Notes for others.

- ECAM: The IMAGE\_REQUEST\_PARM.GROUP\_APPLICABILITY\_FLAG for thumbnails is set to FALSE when it should be TRUE.
- ECAM: The DOWNSAMPLE\_METHOD is incorrect. OFFSET\_MODE\_ID must be used with Table 5-4 to determine the method.
- ECAM: STRIPING\_COUNT and STRIPING\_OVERLAP\_ROWS are incorrect. Striping was not used on the Cruise FSW.

## 15. COREGISTRATION AND LOCALIZATION

Coregistration generally is the process of determining how two observations of the same object relate to each other, in order to better describe or characterize the object. Localization is the process of determining where the rover is in the world. Both share some common characteristics, and are described in this section.

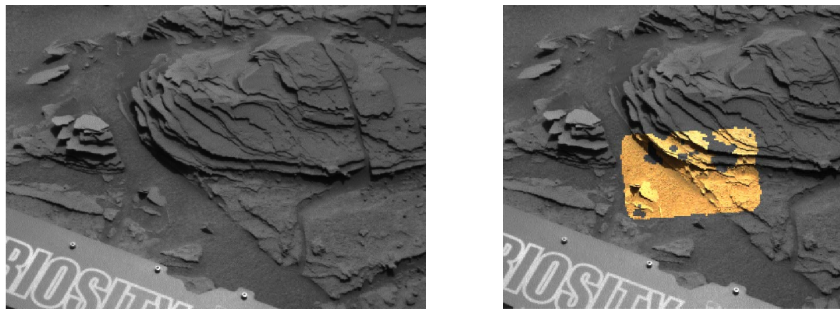
### Image Coregistration

Image coregistration attempts to match pairs of images by matching features within them. The result is that the images can be overlayed on each other in order to see the same object from a different point of view, different camera, or different wavelength (filter).

Mars 2020 implemented a mechanism for systematic coregistration of images across all cameras. The primary use case is to show where PIXL, SHERLOC, or SuperCam spectral observations are in the context of other cameras, in order to better show the scientific context of the observation.

It should be noted that at the present time, coregistration results are not expected to be delivered to PDS – they are an operations product. That could change based on demand or perceived usefulness. They are described here nonetheless because this document serves both operations and archive.

There are three components of the image coregistration system: the pipeline, the CRISP server, and the CRUST server. Clients such as marsviewer use the services provided by these components in order to present coregistration results to the user. Figure 15-1 shows an example of this.



#### 15.1.1 ICM Products

Image coregistration is implemented using the same image correlation process as is used for stereo analysis (Section 8.6). Pairs of images are correlated and a disparity map is created, which indicates, for each pixel in the image, the image coordinates of the matching feature in the other image. These disparity maps are called ICMs (Image Coregistration Maps).

ICMs are exactly the same as disparity maps (DSP et al) – the only difference is in usage. ICMs are created for and used by the coregistration system, while disparity maps are used for stereo analysis. That said, there is crosstalk – implementation of non-simultaneous stereo matching (Section 8.1) is accomplished by mining the coregistration results and repurposing ICMs as disparity maps.

In principle, we want to, within a single site/drive, take every image and coregister it with every other. In practice though, we cut this down somewhat to reduce the workload. Note that ICMs are not generated between different rover positions (site/drive)

The general matching rules are:

- Exclude all video, zstack, depth map, and thumbnail images
- Match only within a single site/drive
- Use only nominally-process images (“\_” for special processing flag)
- Use only non-linearized (raw geometry) images
- Match only to the “best” among “same” images (“same” as in Section 19.2)
- Use the RAS product type
- Processing is symmetric, so if A->B has been done, do not redo B->A
- For ECAMS, only a single tile reconstruction is used – multi-resolution if there is one, the one single-resolution if not.

There are additional camera-specific rules described below. The intent is to make the set of ICMs manageable by discarding those unlikely to provide much use. Of particular note is that image correlation breaks down if the difference in resolution between images is too great. We use as a rule of thumb an 8:1 ratio; if the resolution difference is greater than that, we do not try to correlate. Note however that ICM chaining (Section 15.1.4) can bridge this gap.

Navcam – match against:

- Same-eye Navcam, except itself
- Opposite-eye Navcam, except the nominal stereo partner (covered by standard stereo)
- Front hazcam, both eyes
- Mastcam-Z, both eyes
- Watson images with a focus motor count between 4210 and 12680 (for testbed: 4079 to 12745). This translates to best-focus distance > 0.27m (<12680 for cover open and >4210 for cover closed). At that range Watson is 0.1055 mm/pix, which keeps us within the 8:1 ratio vs. the Navcam 0.64mm/pix at 2m range.

Front Hazcam – match against:

- Navcam, both eyes
- Watson (all – the geometry of the workspace is too complex for a simple resolution test as with the Navcam).

Mastcam-Z – match against:

- Same eye Mastcam-Z, except itself
- Opposite-eye Mastcam-Z, except the nominal stereo partner
- Navcam, both eyes
- Watson
- RMI
- ACI, if Mastcam-Z is > 100mm zoom
- MCC, if Mastcam-Z is > 34mm zoom

Note that only filter 0 participates in coregistration. Video frames, Z-stacks and depth maps do not.

RMI – match against:

- RMI, except itself
- Mastcam-Z, both eyes

Watson – match against:

- Watson, except itself
- Navcam, both eyes, when focus is between 4210 and 12680
- Front hazcam
- Mastcam-Z, both eyes
- ACI
- MCC

ACI – match against:

- ACI, except itself
- Mastcam-Z, when Mastcam-Z  $\geq$  100mm zoom
- Watson
- MCC

MCC – match against:

- MCC, except itself
- Mastcam-Z, when Mastcam-Z  $\geq$  34mm (TBD) zoom
- Watson
- ACI

The other cameras – Rear Hazcam, CacheCam, EDL cameras, LVS Camera, MEDA SkyCam, and the Helicopter cameras – do not participate in coregistration.

### 15.1.2 ICM Filename

The ICM filename is based off the filename of the two images being coregistered. In order to make it easier to deal with the various version-like fields, the last 5 characters (starting at the beginning of the camera-specific field) are stripped off the component filenames, so newer versions don't change the underlying ICM name.

The ICM filename looks like this:

ICM-*<stripped\_filename\_A>*-*<stripped\_filename\_B>*-Jvv.VIC

Where the vv at the end is the ICM version number (not the version for any component products). It is simply incremented each time this ICM is processed; there are no other version-like fields. Note the use of dashes. Dashes are not otherwise allowed within M2020 filenames; their use here thus facilitates pattern matching of the filename to extract the images involved.

The ICM so named is in the geometry of image A, containing coordinates in image B. In disparity map terms, A is the reference image and B is the partner.

For example:



Note that files following the same convention but with “.NO\_OVR” extension may also exist, see next section.

### 15.1.3 Pipeline ICM Generation

Coregistration processing takes place in two phases: check, and process.

Check mode runs the `marsecorr` program with the “-check” parameter. This gives a very quick answer as to whether or not the images might overlap, based solely on their geometry (it does not examine pixels). If the answer is no, no further processing is needed. This helps load-balance the pipeline.

If check mode fails, that means there is no possible overlap between these images. This fact is communicated to the CRISP server by creating an empty file with the ICM filename but with an extension “.NO\_OVR” rather than “.VIC”.

If check mode succeeds, the `marsecorr` program is run again without “-check” in order to create the actual ICM file.

Note that because this is based on image correlation, the ICM will have holes and gores, where correlation did not succeed. This is exacerbated by the sometimes disparate viewing geometries and lighting conditions.

### 15.1.4 CRISP Server

CRISP (Co-Registration of Images Server Protocol) is an operations server that provides coregistration results to clients. A full treatment of the CRISP capabilities is beyond the scope of this document, but highlights are presented here.

CRISP receives ICMs from the pipeline and stores them for use. It supports queries for finding the desired ICM, for mapping coordinates between coregistered images, and for warping the entire image through the ICM to create an overlay.

Crucially, CRISP supports *chaining* of ICMs. If the user wishes to map from image A to C, but an ICM for A->C does not exist, it will look for an image B where A->B and B->C ICMs both exist. If so, the results will be chained together to create a composite A->C ICM. This is particularly useful when there is a large resolution difference, for example an MCC image could be mapped through a Watson and Mastcam-Z in order to get to a Navcam, whereas a direct MCC-Navcam ICM would not be feasible. To be sure, this chaining takes advantage of bilinear interpolation so it has some error associated with it, but it still allows locations to be placed in context.

### 15.1.5 CRUST Server

CRUST (Co-Registration Using Spectral Targets) is an operations server that stores the relationship between spectral observations and the locations of those observations within the context image for the observation.

Each of the spectral instruments (PIXL, SHERLOC, and SuperCam) have a context imager (MCC, ACI, and RMI respectively) that is used to document the spectral observations. A context image is taken for each spectral observation (although many observations may share the same context image). The instrument team determines where in the context image the spectrum was taken. This information is provided to IDS and stored in the CRUST database.

The CRUST database can thus be used to plot the locations of each spectral observation on its context image. Data from CRUST will be made available to PDS.

The real power comes when CRUST is combined with CRISP. CRISP allows the image coordinates from the context image to be translated to other imagers, thus providing the location of the spectral observation within a much larger geological context (the context imagers are all very narrow angle).

## 15.2 Localization

Localization is the process of determining where the rover is by comparing in-situ images to an orbital map of the area. The rover is localized after every drive, and sometimes at mid-drive or penultimate locations (the penultimate provides a view of what will be under the rover once it drives its last segment and stops). Localization is generally expressed in Northing and Easting, which are meters north of the projection parallel and east of the prime meridian. This can be converted to latitude/longitude, but meters are easier to work with in operations. Several location estimates are generated for each location, described below. These are collected and stored in the PLACES database (Section 15.3). Localization uses a "base map", which is a single map covering the entirety of the rover operations area, used consistently for both science and operations purposes. Since localization is based on image analysis, localization is relative to the base map; any projection errors in the map are thus incorporated into localization solutions as well. For this reason, if the user wants to use a different map, they should coregister the official base map (included with the PLACES delivery to PDS) to their map, then translate localized coordinates through that coregistration transform, rather than relying on latitude and longitude directly.

### 15.2.1 Onboard Localization

The rover maintains its own localization onboard based on integration of IMU orientation with counting wheel rotations, which is often improved on by onboard visual odometry. Nevertheless, this is a purely local operation (comparing images to the previous) and does not consider a map, which is required for accurate navigation. The onboard localization is important during operations for commanding, but should not be relied on for any form of navigation. It may be tens of hundreds of meters off due to accumulated drift as the mission proceeds.

### 15.2.2 Auto-Localization

New for Mars 2020, an auto localization program determines where the rover is automatically, as part of the IDS pipeline. It follows the following steps:

- 1) Create an orthomosaic from the rover cameras. Low-pass filter and decimate it to achieve the same resolution as the base map.
- 2) Using an initial estimate of position (derived from the rover's onboard localization), create a subset of the base map containing the area of interest, with some margin.
- 3) Correlate the orthomosaic and base maps using a modified normalized cross correlation. This is a standard correlation algorithm, except it has been modified to account for holes

(areas of no data) in the orthomosaic, which are common due to occlusions. It also weights pixels closer to the center of the mosaic more heavily than those in the distance, as nearby objects are projected more accurately.

- 4) Send the results to PLACES (See 15.3)

### 15.2.3 Manual Localization

Auto localization is performed at every post-drive location. However, there is an opportunity for manual adjustment of the localization results in case autoloco fails or generates bad results. The ROLO (ROver Localization) tool presents the orthomosaic and base maps and allows the user to determine the rover location by a variety of mechanisms. The resulting solution (if any) is also stored in PLACES.

### 15.2.4 Science Localization

SCILO (SCIENCE LOcalization) is an operations server that calculates and stores mapping coordinates (easting, northing, elevation) for every science product generated by the mission, excluding RIMFAX. A full treatment of the SCILO capabilities is beyond the scope of this document, but highlights are presented here.

SCILO uses CRISP and CRUST to find the science product pointing in a NAVCAM image, down to the pixel coordinate(s). The server then uses that pixel pointing information to intersect with either the XYZ (see Section 8.7) or XOZ (see Section 8.7.1) products to get the local-level coordinate triplet for a given rover position (RMC). Then, PLACES is queried to get the rover position in mapping coordinates that is added to the XYZ or XOZ value generating a localized coordinate in the mission mapping coordinate base. The mission basemap uses an Equirectangular projection with standard parallel of 18.4663° N with an IAU2000 spherical radius of 3,396,190 m, henceforth the 'Jezero Crater Projection'. We will store coordinates in easting, northing as well as planetocentric longitude and latitude coordinates with MOLA referenced elevation values. The database will store science data product footprints and center points in GeoJSON format [GeoJSON].

## 15.3 PLACES

PLACES is a web service that manages position and attitude localization solutions from multiple input sources. A solution is a single estimate of position, attitude, or both for a given frame relative to some reference frame. Any given frame may have multiple solutions; a single solution for each frame in the traverse is a view.

The telemetry view, populated by the IDS pipeline, represents the positions and attitudes of each frame along the traverse as reported by the rover. No correction is applied to any solutions in the telemetry view. Should a correction need to be made, a new localization solution is created and applied to a view that inherits from telemetry. This way, the original, uncorrected solution is always available, but users can query for the “most correct” presentation of the traverse.

Two tools write localized solutions to PLACES. Automated localized solutions are submitted to PLACES via the marsautoloco VICAR application, invoked in the IDS pipeline. ROLO is used by tactical planners to manually generate localizations; it is also able to submit to PLACES.

In addition to maintaining multiple representations of the rover traverse, PLACES enables complex queries. The primary query users make to PLACES is for the offset of a point  $\langle x, y \rangle$  relative to

This document has been reviewed and determined not to contain export controlled technical data.

frame A in terms of frame B. This has many uses, such as determining how far away a rock is at a given frame, when the initial measurement was taken at an earlier frame. Another use case is determining the latitude/longitude coordinates of the rover at a given frame. This is done by translating the point  $\langle 0, 0 \rangle$  from the rover to the orbital frame.

The most granular frame that PLACES stores is the rover frame; that is, camera level pointings are not stored. For camera level pointings and adjustments, see 10.4 C-POSE.

## 15.4 C-POSE

C-POSE is a web service for storage, retrieval, ingestion, and search of pointing models. It enables search across different cameras, and allows multiple “opinions” of a given pointing model’s values. Individual “opinions” are stored as solutions. A set of solutions across cameras and RMCs is called a view, and a combination of an RMC and a camera is called a pose.

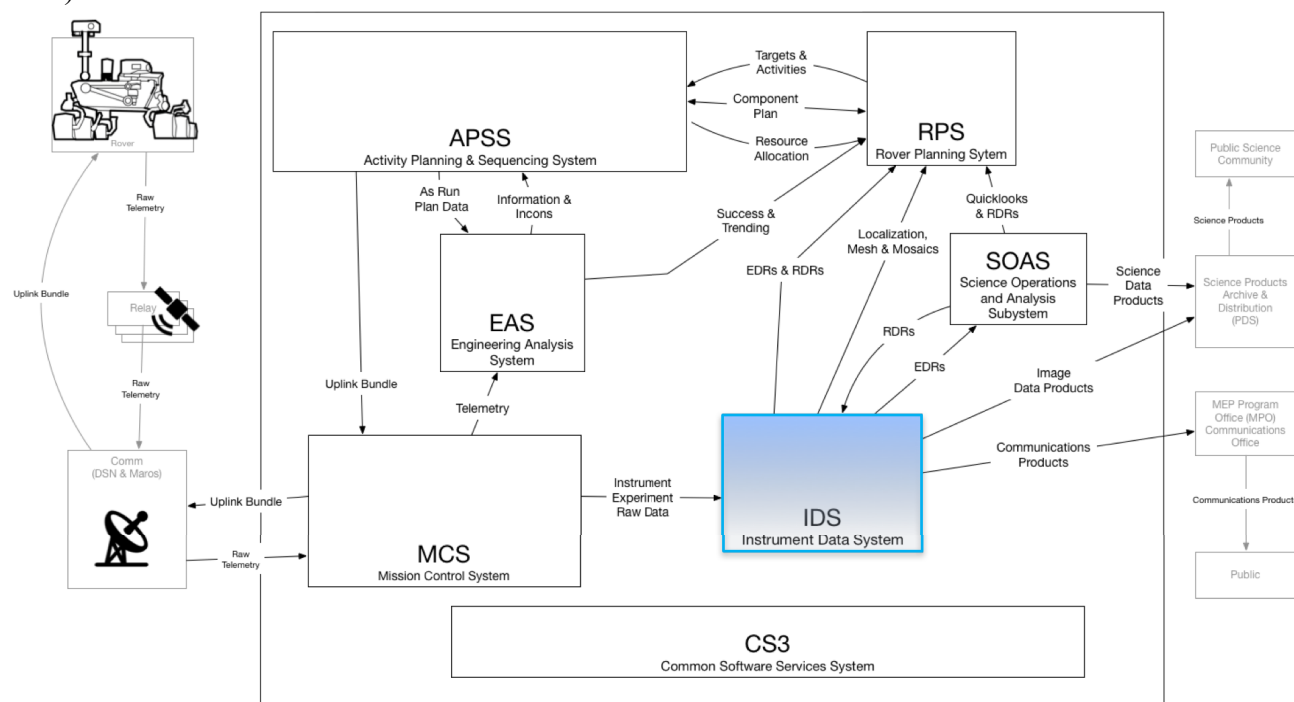
A solution consists of three levels of data, namely pointing models, camera models, and transforms. The latter two pieces of data may be inferred from the former, or they may be supplied explicitly. Multiple different solutions may exist for a given pose, even within a given view. This is achieved by timestamping solutions.

C-POSE does not have a primary query like PLACES does; however, translations across frames may be made by combining C-POSE queries with PLACES’s primary query (see Section 15.3).

## 16. AUTOMATION AND DISTRIBUTION

### 16.1 Cloud Architecture

The Mars 2020 GDS tools and services, with few exceptions, utilize cloud computing and are deployed to and operated using Amazon Web Services (AWS) <https://aws.amazon.com/>. The IDS tools co-exist on the cloud, within an ecosystem of other Mars 2020 subsystem tools (see Figure 16-1).



**Figure 16-1: Mission Operations Architecture**

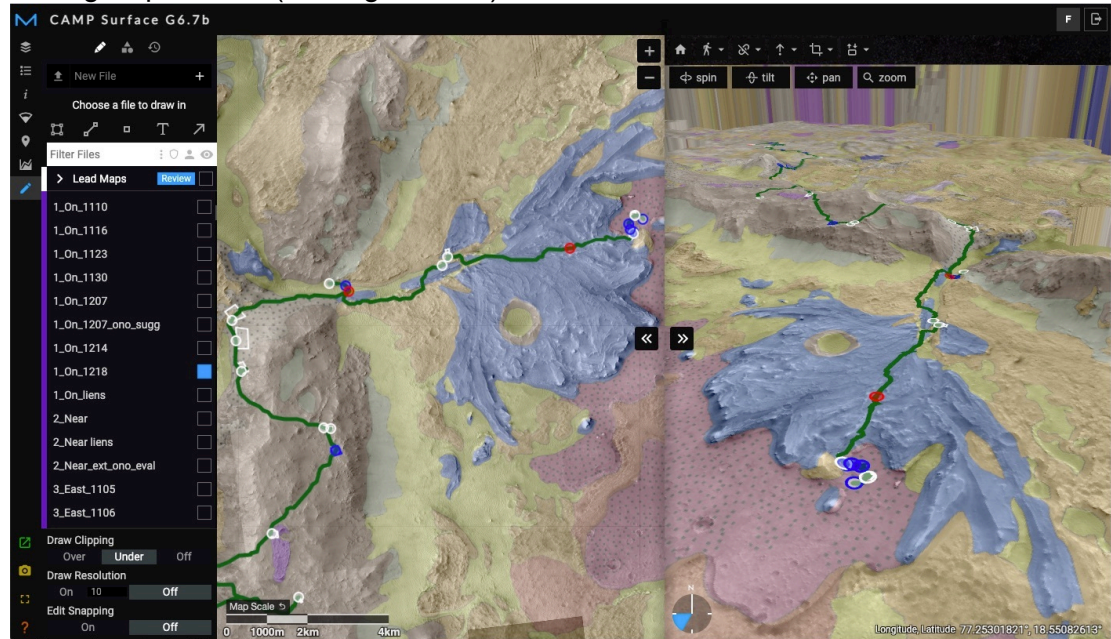
IDS has interfaces with several other subsystems, in order to provide them with data, and programmatic interfaces which expose data. For example, as shown in the diagram above, RPS (Rover Planning System) needs to consume localization solutions, meshes, and mosaics, which their tools require (as integral part of rover planning activities).

The high-level set of IDS tools deployed to the AWS cloud are as follows:

- **IDS Pipeline**
  - Processes raw data from raw data into higher-level data products, and makes these products centrally accessible on the mission data store (S3)
- **PLACES** (Position Localization and Attitude Correction Estimate Storage)
  - A REST-style web service for managing localizations from different sources, and presenting each as a unique view of the rover's traverse. It enables translations across the traverse, allowing targets expressed in a given frame to be understood in the context of another frame.
- **Landform**
  - Generates contextual and tactical 3D meshes used by other subsystems (i.e. RPS)
- **DataDrive**

This document has been reviewed and determined not to contain export controlled technical data.

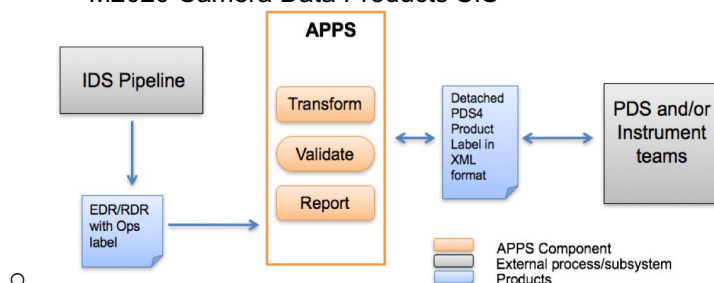
- Provides an interface to search for and download data products in the mission data store.
- **Marsviewer**
  - An interface for viewing and searching for imagery products in the mission data store.
- **CRISP** (Co-Registration of Images Server Protocol)
  - A set of RESTful web services that capture and help identify the features that match among images. The mathematical means to this co-registration of images are inter-camera co-registration map (ICM) and computer graph algorithms.
- **CRUST** (Co-Registration Using Spectral Targets)
  - A database that stores the relationship between spectral observations and the locations of those observations within the context image for the observation.
- **CAMP** (Campaign Analysis Mapping and Planning)
  - A web-based geographical information system (GIS) that provides orbital context and enables science team members to spatially capture interest and science intent for strategic operations (see Figure 16-2).



**Figure 16-2: A screenshot of the Campaign Analysis Mapping and Planning (CAMP) Application**

- **APPS** (AMMOS-PDS Pipeline Service)
  - A software suite that supports PDS4 archive design and production (see Figure 16-3)



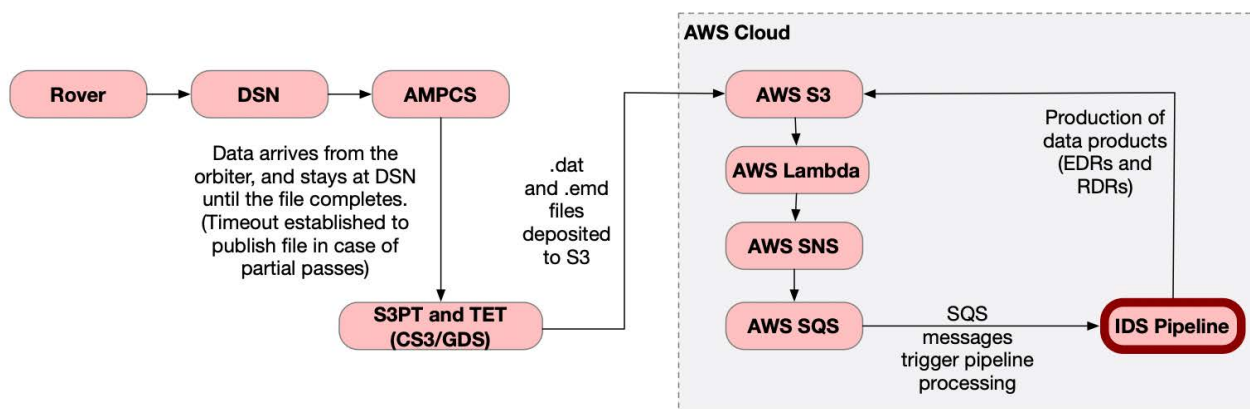


**Figure 16-3: PDS4 Label Generation using APPS**

These tools rely on AWS infrastructure to run. In particular, S3, SQS, SNS, Lambda, and RDS are key infrastructure components.

## 16.2 Mission Data Flow

As mentioned in the Cloud Architecture section, the IDS Pipeline is deployed to the cloud, and is an event-driven system, that's fully automated. Below is a high-level representation of the overall data flow, from the rover, all the way to the IDS Pipeline:



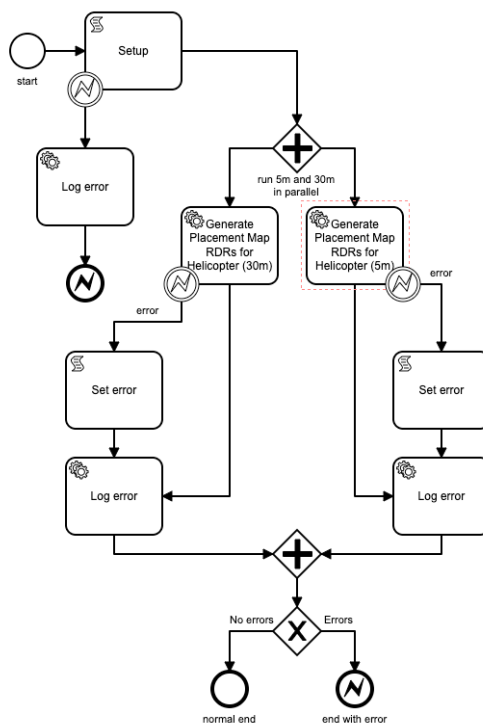
**Figure 16-4: Data Flow diagram from acquisition on the spacecraft to processing by IDS pipeline**

The rover performs data collection from various instruments on the rover, and this data is flowed through different software systems (DSN, AMPCS, S3PT/TET). After passing through these systems, the data arrives on the AWS Cloud – specifically a S3 bucket. From here, some other AWS infrastructure services (Lambda, SNS, SQS) are leveraged to notify the IDS Pipeline there is work to do, and to provide the pipeline with the necessary data products. The pipeline processes the relevant data, and puts newly-created data products back onto S3.

## 16.3 Pipeline Automation

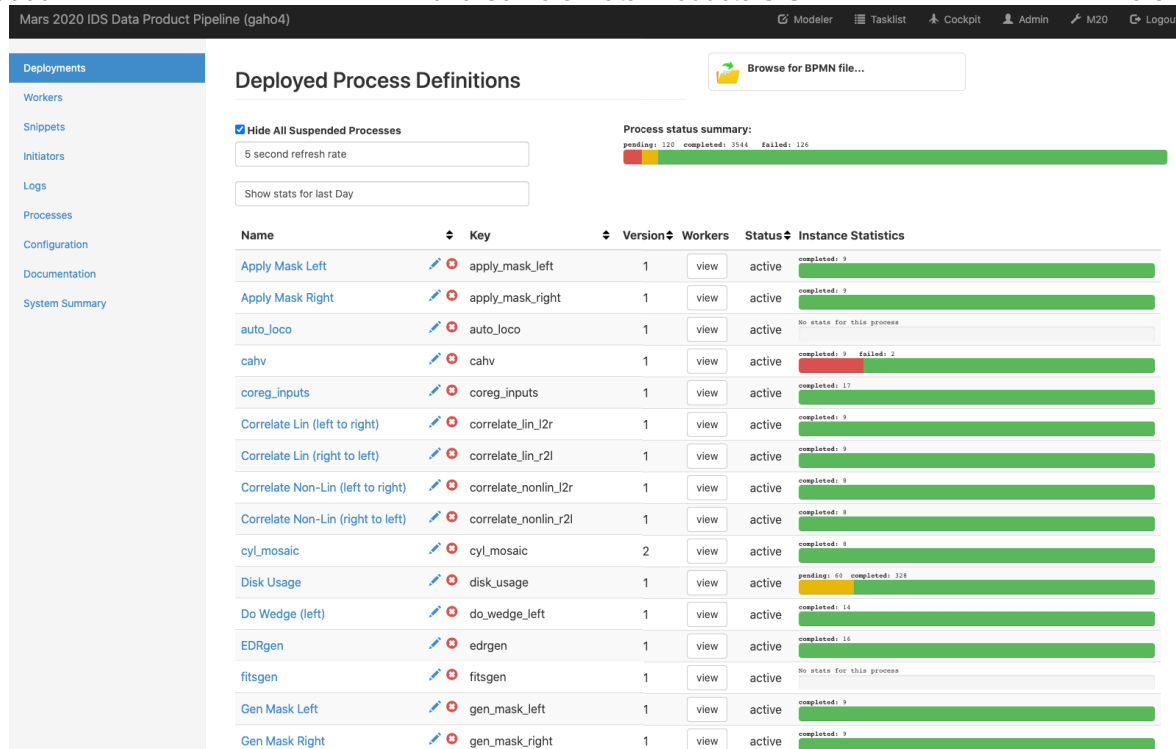
The IDS Pipeline is built as a mission-adaptation, of a Common Workflow Service (CWS) - a core pipeline platform [CWS2017]. CWS is a platform capable of automating science data systems, mission operations processes, and other types of workflows, avoiding disparate workflow implementations. The CWS is a collaborative and standards-based solution using techniques from

the Business Process Management (BPM) discipline, and leverages a standards-based BPMN v2.0 [BPMN], open-source workflow solution, and a web-based platform. By leveraging the BPMN 2.0 standard, processes are easily understood and presentable to all stakeholders (from developers to managers). Figure 16-5 shows an example of a typical processing flow in the IDS Pipeline. Distribution of workload across multiple worker machines, web-based monitoring (see Figure 16-6), and support for user/custom tasks come out-of-the-box with CWS, and are used in the IDS Pipeline to allow for accomplishing the data processing for a Mars 2020 data pass, within the tactical timeframe.



**Figure 16-5: Typical BPMN Diagram for an IDS Pipeline product generation Flow**





**Figure 16-6: The IDS Pipeline Console web application**

Appendix D contains a diagram showing the full data flow of the IDS pipeline.

## 16.4 Data Distribution

All files are distributed, saved and viewed via the Mars2020 OCS (Operations Cloud Store). OCS is a file storage system developed on top of S3 with metadata management capabilities such as specifying custom metadata fields and full text search through metadata fields. Data products in OCS are grouped into different packages and is categorized into different object types. A package is essentially a top level folders in OCS and is mapped one to one with a S3 bucket. An object type is metadata that categorizes similar data products into one category. For example, all EDR and RDR data products would be categorized into a single object type.

Ingestion of data products into OCS can be done in multiple ways. The first couple of methods are done via the OCS CLI, OCS JS client library and the OCS Python client library. Additionally, if a user would like to view, save and/or search data products via a web browser, he/she can utilize DataDrive. DataDrive is a web application for OCS and is capable to providing features of OCS but on a web browser. Lastly, if the user has access to AWS S3 buckets directly, he/she can upload data products directly into a S3 buckets that has been mapped to an OCS package give that OCS Ingester has been enabled for said bucket. OCS Ingester is a service that listens for new data products uploaded into a S3 bucket and automatically registers them into OCS.

## 17. PRODUCT TYPE CODES

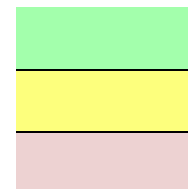
The image RDR data products covered by this SIS are listed in Table 17-2 and Table 17-3 below. Products listed as “1-3” bands are generally 1-band products but could have 3 bands RGB bands for certain camera modes. Table 17-1 defines the color codes used in Table 17-2.

**Table 17-1 – Product Type color codes**

Primary to End-User: most important, most used, most popular products

Secondary: intermediate products, or final product not commonly used

Special: products generated outside of the pipeline as a special request



**Table 17-2 – M2020 Product Type Codes**

Data Product	Type	PDS Proc. Level	Derived Image Type
<b>EDR</b>			
Original Image Product, Possibly Companded (8->12bit)	<b>ECM</b>	Raw	N/A
Original JPEG as received from the rover	<b>EJP</b>	Raw	N/A
Single frame of a Video Sequence (possibly companded)	<b>ECV</b>	Raw	N/A
Z-stack frame (possibly companded)	<b>ECZ</b>	Raw	N/A
Depth Map	<b>EDM</b>	Raw	N/A
Recovered Product (possibly companded)	<b>ECR</b>	Raw	N/A
Reference Pixel	<b>ERP</b>	Raw	N/A
Row Summed	<b>ERS</b>	Raw	N/A
Column Summed	<b>ECS</b>	Raw	N/A
Histogram	<b>EHG</b>	Raw	N/A
Audio	<b>EAU</b>	Raw	N/A
Decompanded Image Product	<b>EDR</b>	Partial	IMAGE
Single frame of a Video Sequence (de-companded)	<b>EVD</b>	Partial	IMAGE
Z-stack frame (de-companded)	<b>EZS</b>	Partial	IMAGE
Recovered Product (de-companded)	<b>ERD</b>	Partial	IMAGE
De-Bayered (de-mosaicked)	<b>EBY</b>	Partial	IMAGE
<b>FDR</b>			
Base image from which all RDRs are derived	<b>FDR</b>	Partial	IMAGE
Tile Data Record	<b>TDR</b>	Partial	IMAGE
Audio Reconstructed	<b>FAU</b>	Partial	N/A
<b>Radiometric</b>			
Rad-corrected IOF radiance factor, float	<b>IOF</b>	Calibrated	IMAGE
Rad-corrected IOF radiance factor, integer	<b>IOI</b>	Calibrated	IMAGE
Rad-corrected absolute radiance units, float	<b>RAF</b>	Calibrated	IMAGE
Rad-corrected absolute radiance units, 15-bit integer static scale	<b>RAD</b>	Calibrated	IMAGE

This document has been reviewed and determined not to contain export controlled technical data.

M2020 Camera Data Products C10							Version 0.1
Data Product	Type				PDS Proc. Level	Derived Image Type	
Rad-corrected absolute radiance units, 12-bit integer static scale	RAS				Calibrated	IMAGE	
Rad-corrected absolute radiance units, 15-bit int dynamic scale	RAY				Calibrated	IMAGE	
Rad-corrected radiance (RAD), masked	RDM				Calibrated	IMAGE	
Rad-corrected for Instrument Effects only, integer DN	RIE				Calibrated	IMAGE	
Rad-corrected for Instrument Effects only, float	RIF				Calibrated	IMAGE	
Rad-corrected scaled radiance (RAS), masked	RSM				Calibrated	IMAGE	
Gamma-corrected radiance, no longer calibrated units, byte	RAG				Derived	IMAGE	
Zenith-scaled radiance, float	RZF				Calibrated	IMAGE	
Zenith-scaled radiance, 15-bit integer static scale	RZD				Calibrated	IMAGE	
Zenith-scaled radiance, 12-bit integer static scale	RZS				Calibrated	IMAGE	
Zenith-scaled radiance, 15-bit integer dynamic scale	RZY				Calibrated	IMAGE	
Color							
Exposure Time Compensated Color	Cxx						
Raw DN Color		Dxx					
Zenith Scaled Color			Zxx				
Masked Color				Mxx			
Color, No Illuminant, 15-bit scaled integer, linear space	CNR	DNR	ZNR	MNR	Derived	IMAGE	
Color, No Illuminant, 12-bit scaled integer, linear space	CNS	DNS	ZNS	MNS	Derived	IMAGE	
Color, No Illuminant, Float, linear space	CNF	DNF	ZNF	MNF	Derived	IMAGE	
Color, No Illuminant, Byte, linear space	CNB	DNB	ZNB	MNB	Derived	IMAGE	
Color, No Illuminant, Byte, Gamma-corrected	CNG	DNG	ZNG	MNG	Derived	IMAGE	
Color, Standard Illuminant, 15-bit scaled integer, linear space	CSD	DSD	ZSD	MSD	Derived	IMAGE	
Color, Standard Illuminant, 12-bit scaled integer, linear space	CSS	DSS	ZSS	MSS	Derived	IMAGE	
Color, Standard Illuminant, Float, linear space	CSF	DSF	ZSF	MSF	Derived	IMAGE	
Color, Standard Illuminant, Byte, linear space	CSB	DSB	ZSB	MSB	Derived	IMAGE	
Color, Standard Illuminant, Byte, Gamma-corrected	CSG	DSG	ZSG	MSG	Derived	IMAGE	
Color, White Bal, Mars Illuminant, 15-bit scaled int, linear space	CPD	DPD	ZPD	MPD	Derived	IMAGE	
Color, White Bal, Mars Illuminant, 12-bit scaled int, linear space	CPS	DPS	ZPS	MPS	Derived	IMAGE	
Color, White Balanced, Mars Illuminant, Float, linear space	CPF	DPF	ZPF	MPF	Derived	IMAGE	
Color, White Balanced, Mars Illuminant, Byte, linear space	CPB	DPB	ZPB	MPB	Derived	IMAGE	
Color, White Balanced, Mars Illuminant, Byte, Gamma-corrected	CPG	DPG	ZPG	MPG	Derived	IMAGE	
Color, White Balanced, 15-bit scaled integer, linear space	CWD	DWD	ZWD	MWD	Derived	IMAGE	
Color, White Balanced, 12-bit scaled integer, linear space	CWS	DWS	ZWS	MWS	Derived	IMAGE	
Color, White Balanced, Float, linear space	CWF	DWF	ZWF	MWF	Derived	IMAGE	
Color, White Balanced, Byte, linear space	CWB	DWB	ZWB	MWB	Derived	IMAGE	
Color, White Balanced, Byte, Gamma-corrected	CWG	DWG	ZWG	MWG	Derived	IMAGE	
Correlation							
Stereo Delta Disparity (2-band, true disparity offset)	DDD				Derived	DISP_DELTA_MAP	
Stereo Delta Disparity Line (single-band)	DDL				Derived	DISP_DELTA_LINE_MAP	
Stereo Delta Disparity Sample (single-band)	DDS				Derived	DISP_DELTA_SAMPLE_MAP	
Stereo Disparity Error Metric	DSE				Derived	DISPARITY_ERROR_MAP	
Stereo Disparity Final	DSP				Derived	DISPARITY_MAP	
Stereo Disparity Raw	DSR				Derived	DISPARITY_MAP	
Stereo Disparity Mask File	MDS				Derived	MASK	

This document has been reviewed and determined not to contain export controlled technical data.

Data Product	Type	PDS Proc. Level	Derived Image Type
Stereo First-stage Disparity Final	<b>DFF</b>	Derived	DISPARITY_MAP
Stereo First-stage Disparity Line	<b>DFL</b>	Derived	DISPARITY_LINE_MAP
Stereo First-stage Disparity Sample	<b>DFS</b>	Derived	DISPARITY_SAMPLE_MAP
Correlation Coefficients, First-stage	<b>COF</b>	Derived	DISPARITY_COEFS_MAP
Correlation Coefficients, Second-stage	<b>COE</b>	Derived	DISPARITY_COEFS_MAP
<b>XYZ</b>			
XYZ expressed in Site frame	<b>XYZ</b>	Calibrated	XYZ_MAP
XYZ expressed in Site frame, Masked	<b>XYM</b>	Calibrated	XYZ_MAP
XYZ expressed in Rover Nav frame	<b>XYR</b>	Calibrated	XYZ_MAP
XYZ expressed in Rover, Masked	<b>XRM</b>	Calibrated	XYZ_MAP
XYZ Error Metric	<b>XYE</b>	Calibrated	XYZ_ERROR_MAP
XYZ Mask File	<b>MXY</b>	Calibrated	MASK
XYZ Filled	<b>XYF</b>	Derived	XYZ_MAP
XYZ with Overlay	<b>XYO</b>	Derived	IMAGE
XYZ from disparity raw expressed in Site frame	<b>XRZ</b>	Derived	XYZ_MAP
XYZ from disparity raw expressed in Rover Nav frame	<b>XRR</b>	Derived	XYZ_MAP
XYZ X-band	<b>XXX</b>	Calibrated	X_MAP
XYZ Y-band	<b>YYY</b>	Calibrated	Y_MAP
XYZ Z-band	<b>ZZZ</b>	Calibrated	Z_MAP
Digital Elevation Model. Like ZZZ but positive up.	<b>DEM</b>	Calibrated	ELEVATION_MAP
Orbital XYZ	<b>XOZ</b>	Calibrated	XYZ_MAP
XYZ X-band Filled	<b>XXF</b>	Derived	X_MAP
XYZ Y-band Filled	<b>YYF</b>	Derived	Y_MAP
XYZ Z-band Filled	<b>ZZF</b>	Derived	Z_MAP
Instrument Z value for Helicopter	<b>ZIH</b>	Derived	Z_MAP
<b>Range</b>			
Range Error Metric	<b>RNE</b>	Calibrated	RANGE_ERROR_MAP
Range Filled	<b>RNF</b>	Derived	RANGE_MAP
Range from Camera	<b>RNG</b>	Calibrated	RANGE_MAP
Range from Camera, Masked	<b>RNM</b>	Calibrated	RANGE_MAP
Range from Rover Origin	<b>RNR</b>	Calibrated	RANGE_MAP
Range with Overlay	<b>RNO</b>	Derived	IMAGE
<b>Surface Normal</b>			
Surface Normal (UVW) Filled	<b>UVF</b>	Derived	UVW_MAP
Surface Normal (UVW) with Overlay	<b>UVO</b>	Derived	IMAGE
Surface Normal (UVW) Projected onto Plane	<b>UVP</b>	Derived	UVW_MAP
Surface Normal (UVW) for Slope computations	<b>UVS</b>	Calibrated	UVW_MAP
Surface Normal (UVW) Angle (Theta) between Normal and Plane	<b>UVT</b>	Derived	ANGLE_MAP
Surface Normal (UVW)	<b>UVW</b>	Calibrated	UVW_MAP
Surface Normal (UVW) U-band	<b>UUU</b>	Calibrated	U_MAP
Surface Normal (UVW) V-band	<b>VVV</b>	Calibrated	V_MAP
Surface Normal (UVW) W-band	<b>WWW</b>	Calibrated	W_MAP
Surface Normal (UVW) U-band Filled	<b>UUF</b>	Derived	U_MAP

Data Product	Type	PDS Proc. Level	Derived Image Type
Surface Normal (UVW) V-band Filled	VVF	Derived	V_MAP
Surface Normal (UVW) W-band Filled	WWF	Derived	W_MAP
Instrument Normal for Helicopter	UIH	Derived	????
Tilt for Helicopter	TFH	Derived	TILT_MAP
Slope			
Solar Energy	SEN	Derived	SOLAR_ENERGY_MAP
Slope Heading	SHD	Derived	SLOPE_HEADING_MAP
Slope Heading with Overlay	SHO	Derived	IMAGE
Slope with Overlay	SLO	Derived	IMAGE
Slope	SLP	Derived	SLOPE_MAP
Slope Magnitude	SMG	Derived	SLOPE_MAGNITUDE_MAP
Slope Magnitude with Overlay	SMO	Derived	IMAGE
Slope Northerly Tilt with Overlay	SNO	Derived	IMAGE
Slope Northerly Tilt	SNT	Derived	NORTHERLY_TILT_MAP
Slope Radial Direction	SRD	Derived	RADIAL_SLOPE_MAP
Reachability			
Arm Reachability, Masked	ARK	Derived	REACHABILITY_MAP
Arm Reachability	ARM	Derived	REACHABILITY_MAP
Arm Reachability with overlay	ARO	Derived	IMAGE
Arm Reachability Mask File	MAR	Derived	MASK
Roughness			
Surface Roughness (general, not instrument-specific)	RUF	Derived	ROUGHNESS_MAP
Surface Roughness (for Drill Stabilizer and Coring)	RUS	Derived	ROUGHNESS_MAP
Surface Curvature for Drill	RUC	Derived	ROUGHNESS_MAP
Surface Roughness for Helicopter	RUH	Derived	ROUGHNESS_MAP
Goodness			
Goodness (Overall for Natural Surface Tools)	GUN	Derived	GOODNESS_MAP
Goodness (Overall for Abrading Tools)	GUA	Derived	GOODNESS_MAP
Goodness (Overall for Coring Tools)	GUC	Derived	GOODNESS_MAP
Goodness (For Reachability of Natural Surface Tools)	GRN	Derived	GOODNESS_MAP
Goodness (For Reachability of Abrading Tools)	GRA	Derived	GOODNESS_MAP
Goodness (For Reachability of Coring Tools)	GRC	Derived	GOODNESS_MAP
Goodness (Combined for Surface Roughness)	GSR	Derived	GOODNESS_MAP
Goodness (Helicopter)	GUH	Derived	GOODNESS_MAP
Slope Goodness for Helicopter	SLH	Derived	GOODNESS_MAP
Miscellaneous			
Image Mask File	MSK	Derived	MASK
Terrain Class Confidence	TEN	Derived	TERRAIN_CLASSIFICATION_MAP
Terrain Class	TER	Derived	TERRAIN_CLASSIFICATION_MAP
Incidence, Emission, Phase angles Filled	IEF	Derived	IEP_MAP
Incidence, Emission, Phase angles	IEP	Derived	IEP_MAP
Index Depth Map	IDM	Derived	IDM_MAP
Mosaic Input Index File	IDX	Derived	IDX_MAP

This document has been reviewed and determined not to contain export controlled technical data.

Data Product	Type	PDS Proc. Level	Derived Image Type
Mosaic Input Coregistration Map	ICM	Derived	ICM_MAP

**Table 17-3 – M2020 Product Type Codes, Sorted Alphabetically**

Code	Type	Bands	Section	Category
ARK	short	<u>6</u>	<u>10.2</u>	Reachability
ARM	short	<u>6</u>	<u>10.2</u>	Reachability
ARO	?	<u>2</u>	<u>10.2</u>	Reachability
CNB	byte	<u>3</u>	<u>7.2</u>	Color, ETC
CNF	float	<u>3</u>	<u>7.2</u>	Color, ETC
CNG	byte	<u>3</u>	<u>7.2</u>	Color, ETC
CNR	short	<u>3</u>	<u>7.2</u>	Color, ETC
CNS	short	<u>3</u>	<u>7.2</u>	Color, ETC
COE	float	<u>6</u>	<u>8.5</u>	Correlation
COF	float	<u>6</u>	<u>8.5</u>	Correlation
CPB	byte	<u>3</u>	<u>7.2</u>	Color, ETC
CPD	short	<u>3</u>	<u>7.2</u>	Color, ETC
CPF	float	<u>3</u>	<u>7.2</u>	Color, ETC
CPG	byte	<u>3</u>	<u>7.2</u>	Color, ETC
CPS	short	<u>3</u>	<u>7.2</u>	Color, ETC
CSB	byte	<u>3</u>	<u>7.2</u>	Color, ETC
CSD	short	<u>3</u>	<u>7.2</u>	Color, ETC
CSF	float	<u>3</u>	<u>7.2</u>	Color, ETC
CSG	byte	<u>3</u>	<u>7.2</u>	Color, ETC
CSS	short	<u>3</u>	<u>7.2</u>	Color, ETC
CWB	byte	<u>3</u>	<u>7.2</u>	Color, ETC
CWD	short	<u>3</u>	<u>7.2</u>	Color, ETC
CWF	float	<u>3</u>	<u>7.2</u>	Color, ETC
CWG	byte	<u>3</u>	<u>7.2</u>	Color, ETC
CWS	short	<u>3</u>	<u>7.2</u>	Color, ETC
DDD	float	<u>2</u>	<u>8.6</u>	Correlation
DDL	float	<u>1</u>	<u>8.6</u>	Correlation
DDS	float	<u>1</u>	<u>8.6</u>	Correlation
DEM	float	<u>1</u>	<u>8.7</u>	XYZ
DFE	float	<u>2</u>	<u>8.6</u>	Correlation
DFL	float	<u>1</u>	<u>8.6</u>	Correlation
DFS	float	<u>1</u>	<u>8.6</u>	Correlation
DNB	byte	<u>3</u>	<u>7.2</u>	Color, Raw DN
DNF	float	<u>3</u>	<u>7.2</u>	Color, Raw DN
DNG	byte	<u>3</u>	<u>7.2</u>	Color, Raw DN
DNR	short	<u>3</u>	<u>7.2</u>	Color, Raw DN
DNS	short	<u>3</u>	<u>7.2</u>	Color, Raw DN
DPB	byte	<u>3</u>	<u>7.2</u>	Color, Raw DN
DPD	short	<u>3</u>	<u>7.2</u>	Color, Raw DN
DPF	float	<u>3</u>	<u>7.2</u>	Color, Raw DN
DPG	byte	<u>3</u>	<u>7.2</u>	Color, Raw DN

Code	Type	Bands	Section	Category
DPS	short	<u>3</u>	<u>7.2</u>	Color, Raw DN
DSB	byte	<u>3</u>	<u>7.2</u>	Color, Raw DN
DSD	short	<u>3</u>	<u>7.2</u>	Color, Raw DN
DSE	float	<u>4</u>	<u>8.7.1</u>	Correlation
DSF	float	<u>3</u>	<u>7.2</u>	Color, Raw DN
DSG	byte	<u>3</u>	<u>7.2</u>	Color, Raw DN
DSL	float	<u>1</u>	<u>8.6</u>	Correlation
DSP	float	<u>2</u>	<u>8.6</u>	Correlation
DSR	float	<u>2</u>	<u>8.6</u>	Correlation
DSS	short	<u>3</u>	<u>7.2</u>	Color, Raw DN
DSS	float	<u>1</u>	<u>8.6</u>	Correlation
DWB	byte	<u>3</u>	<u>7.2</u>	Color, Raw DN
DWD	short	<u>3</u>	<u>7.2</u>	Color, Raw DN
DWF	float	<u>3</u>	<u>7.2</u>	Color, Raw DN
DWG	byte	<u>3</u>	<u>7.2</u>	Color, Raw DN
DWS	short	<u>3</u>	<u>7.2</u>	Color, Raw DN
EAU	byte	<u>n/a</u>	<u>5.9.2</u>	EDR
EBY	short	<u>1.3</u>	<u>5.2.3</u>	EDR
ECM	byte,short	<u>1.3</u>	<u>5.2.1</u>	EDR
ECR	byte,short	<u>1.3</u>	<u>5.2.1</u>	EDR
ECV	byte,short	<u>1.3</u>	<u>5.2.1</u>	EDR
ECZ	byte,short	<u>1.3</u>	<u>5.2.1</u>	EDR
EDM	byte	<u>1</u>	<u>5.3.3</u>	EDR
EDR	short	<u>1.3</u>	<u>5.2.2</u>	EDR
EJP	*	<u>*</u>	<u>5.3.6</u>	EDR
ERD	short	<u>1.3</u>	<u>5.2.2</u>	EDR
ERP	?	<u>?</u>	<u>5.7.3</u>	EDR
EVD	short	<u>1.3</u>	<u>5.2.2</u>	EDR
EZS	short	<u>1.3</u>	<u>5.2.2</u>	EDR
FAU	byte	<u>n/a</u>	<u>6.4</u>	FDR
FDR	short	<u>1.3</u>	<u>6</u>	FDR
GRA	byte	<u>1.2</u>	<u>10.3</u>	Goodness
GRC	byte	<u>1.2</u>	<u>10.3</u>	Goodness
GRN	byte	<u>1.2</u>	<u>10.3</u>	Goodness
GSR	byte	<u>1</u>	<u>10.3</u>	Goodness
GUA	byte	<u>1</u>	<u>10.3</u>	Goodness
GUC	byte	<u>1</u>	<u>10.3</u>	Goodness
GUH	byte	<u>1</u>	<u>14.2.2.3</u>	Goodness
GUN	byte	<u>1</u>	<u>10.3</u>	Goodness
ICM	float	<u>2</u>	<u>12.9</u>	Miscellaneous
IDM	byte	<u>1</u>	<u>6.2.1</u>	Miscellaneous

This document has been reviewed and determined not to contain export controlled technical data.

Code	Type	Bands	Section	Category
IDX	short	<u>1</u>	<u>12.9</u>	Miscellaneous
IEF	float	<u>3</u>		Miscellaneous
IEP	float	<u>3</u>		Miscellaneous
IOF	float	<u>1,3</u>	<u>7.1</u>	Radiometric
IOI	short	<u>1,3</u>	<u>7.1</u>	Radiometric
MAR	byte	<u>1</u>	<u>10.2</u>	Reachability
MDS	byte	<u>1</u>	<u>8.6</u>	Correlation
MNB	byte	<u>3</u>	<u>7.2</u>	Color, Masked
MNF	float	<u>3</u>	<u>7.2</u>	Color, Masked
MNG	byte	<u>3</u>	<u>7.2</u>	Color, Masked
MNR	short	<u>3</u>	<u>7.2</u>	Color, Masked
MNS	short	<u>3</u>	<u>7.2</u>	Color, Masked
MPB	byte	<u>3</u>	<u>7.2</u>	Color, Masked
MPD	short	<u>3</u>	<u>7.2</u>	Color, Masked
MPF	float	<u>3</u>	<u>7.2</u>	Color, Masked
MPG	byte	<u>3</u>	<u>7.2</u>	Color, Masked
MPS	short	<u>3</u>	<u>7.2</u>	Color, Masked
MSB	byte	<u>3</u>	<u>7.2</u>	Color, Masked
MSD	short	<u>3</u>	<u>7.2</u>	Color, Masked
MSF	float	<u>3</u>	<u>7.2</u>	Color, Masked
MSG	byte	<u>3</u>	<u>7.2</u>	Color, Masked
MSK	byte			Miscellaneous
MSS	short	<u>3</u>	<u>7.2</u>	Color, Masked
MWB	byte	<u>3</u>	<u>7.2</u>	Color, Masked
MWD	short	<u>3</u>	<u>7.2</u>	Color, Masked
MWF	float	<u>3</u>	<u>7.2</u>	Color, Masked
MWG	byte	<u>3</u>	<u>7.2</u>	Color, Masked
MWS	short	<u>3</u>	<u>7.2</u>	Color, Masked
MXV	byte	<u>3</u>	<u>8.7</u>	XYZ
RAD	short	<u>1,3</u>	<u>7.1</u>	Radiometric
RAF	float	<u>1,3</u>	<u>7.1</u>	Radiometric
RAG	byte	<u>1,3</u>	<u>7.1</u>	Radiometric
RAS	short	<u>1,3</u>	<u>7.1</u>	Radiometric
RAY	short	<u>1,3</u>	<u>7.1</u>	Radiometric
RDM	short	<u>1,3</u>	<u>7.1</u>	Radiometric
RIE	short	<u>1,3</u>	<u>7.1</u>	Radiometric
RIF	float	<u>1,3</u>	<u>7.1</u>	Radiometric
RNE	float	<u>3</u>	<u>8.7.1</u>	Range
RNF	float	<u>1</u>		Range
RNG	float	<u>1</u>		Range
RNM	float	<u>1</u>		Range
RNO	?	<u>2</u>		Range
RNR	float	<u>1</u>		Range
RSM	short	<u>1,3</u>	<u>7.1</u>	Radiometric
RUC	float	<u>3</u>	<u>10.1</u>	Roughness
RUF	float	<u>3</u>	<u>10.1</u>	Roughness

Code	Type	Bands	Section	Category
RUH	float	<u>3</u>	<u>14.2.2.2</u>	Roughness
RUS	float	<u>3</u>	<u>10.1</u>	Roughness
RZD	short	<u>1,3</u>	<u>7.1.5</u>	Radiometric
RZF	float	<u>1,3</u>	<u>7.1.5</u>	Radiometric
RZS	short	<u>1,3</u>	<u>7.1.5</u>	Radiometric
RZY	short	<u>1,3</u>	<u>7.1.5</u>	Radiometric
SEN	float	<u>1</u>		Slope
SHD	float	<u>1</u>	<u>8.13.2</u>	Slope
SHO	?	<u>2</u>		Slope
SLH	byte	<u>1</u>	<u>14.2.2.3</u>	Goodness
SLO	?	<u>2</u>		Slope
SLP	float	<u>1</u>	<u>8.13.1</u>	Slope
SMG	float	<u>1</u>	<u>8.13.4</u>	Slope
SMO	?	<u>2</u>		Slope
SNO	?	<u>2</u>		Slope
SNT	float	<u>1</u>	<u>8.13.5</u>	Slope
SRD	float	<u>1</u>	<u>8.13.3</u>	Slope
TDR	short	<u>1,3</u>	<u>6.2.2</u>	FDR
TEN	float	<u>2</u>		Miscellaneous
TER	?	<u>2</u>		Miscellaneous
TFH	float	<u>3</u>	<u>14.2.2.2</u>	Surface Normal
UIH	float	<u>3</u>	<u>14.2.2.2</u>	Surface Normal
UUF	float	<u>1</u>	<u>8.10</u>	Surface Normal
UUU	float	<u>1</u>	<u>8.10</u>	Surface Normal
UVF	float	<u>3</u>	<u>8.10</u>	Surface Normal
UVO	?	<u>2</u>		Surface Normal
UVP	float	<u>1??</u>	<u>8.10</u>	Surface Normal
UVS	float	<u>3</u>	<u>8.10</u>	Surface Normal
UVT	float	<u>1</u>	<u>8.10</u>	Surface Normal
UVW	float	<u>3</u>	<u>8.10</u>	Surface Normal
VVF	float	<u>1</u>	<u>8.10</u>	Surface Normal
VVV	float	<u>1</u>	<u>8.10</u>	Surface Normal
WWF	float	<u>1</u>	<u>8.10</u>	Surface Normal
WWW	float	<u>1</u>	<u>8.10</u>	Surface Normal
XOZ	float	<u>3</u>	<u>8.7.1</u>	XYZ
XRM	float	<u>3</u>	<u>8.7</u>	XYZ
XRR	float	<u>3</u>	<u>8.7</u>	XYZ
XRZ	float	<u>3</u>	<u>8.7</u>	XYZ
XXF	float	<u>1</u>	<u>8.7</u>	XYZ
XXX	float	<u>1</u>	<u>8.7</u>	XYZ
XYE	float	<u>3</u>	<u>8.7.1</u>	XYZ
XYF	float	<u>3</u>	<u>8.7</u>	XYZ
XYM	float	<u>3</u>	<u>8.7</u>	XYZ
XYO	?	<u>2</u>	<u>8.7</u>	XYZ
XYR	float	<u>3</u>	<u>8.7</u>	XYZ
XYZ	float	<u>3</u>	<u>8.7</u>	XYZ

This document has been reviewed and determined not to contain export controlled technical data.

Code	Type	Bands	Section	Category
<b>YYF</b>	float	<u>1</u>	<u>8.7</u>	XYZ
<b>YYY</b>	float	<u>1</u>	<u>8.7</u>	XYZ
<b>ZIH</b>	float	<u>3</u>	<u>14.2.2.2</u>	XYZ
<b>ZNB</b>	byte	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZNF</b>	float	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZNG</b>	byte	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZNR</b>	short	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZNS</b>	short	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZPB</b>	byte	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZPD</b>	short	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZPF</b>	float	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZPG</b>	byte	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZPS</b>	short	<u>3</u>	<u>7.2</u>	Color, Zenith

Code	Type	Bands	Section	Category
<b>ZSB</b>	byte	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZSD</b>	short	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZSF</b>	float	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZSG</b>	byte	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZSS</b>	short	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZWB</b>	byte	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZWD</b>	short	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZWF</b>	float	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZWG</b>	byte	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZWS</b>	short	<u>3</u>	<u>7.2</u>	Color, Zenith
<b>ZZF</b>	float	<u>1</u>	<u>8.7</u>	XYZ
<b>ZZZ</b>	float	<u>1</u>	<u>8.7</u>	XYZ



## 18. FILE NAMING STANDARDS

The file naming scheme has been relieved of the string length constraint imposed by the Level II 36.3 filename standard approved by PDS in 2009. PDS has migrated to a new standard that is PDS-4, and filenames are essentially limited to a character string length of 255. This is a change from the 36.3 convention that MSL was constrained to using. Use of three- character extensions, such as “.IMG” for image EDRs and RDRs and “.DAT” for spectrum EDRs and state-of-health EDRs, is consistent with the PDS standard.

There are three file naming schemes adapted for the M2020 image and non-image data products. The first applies to the EDR data product and all Single-frame RDR data products. The second applies to all Mosaic RDR data products. The third applies to Terrain products.

The primary attributes of the filename nomenclature are:

- a) Uniqueness - It must be unique unto itself without the file system’s directory path. This Protects against product overwrite as files are copied/moved within the file system and external to the file system, if managed correctly.
- b) Metadata - It should be comprised of metadata fields that keep file bookkeeping and sorting intuitive to the human user. Even though autonomous file processing will be managed via databases, there will always be human-in-the-loop that puts a premium on filename intuition. Secondly, the metadata fields should be smartly selected based on their value to ground processing tools, as it is less CPU-intensive to extract information from the filename than from the label.

NOTE: Most metadata information in the filename is also in the product label.

The metadata fields have been selected based on MSL, MER, and PHX lessons learned. In general, the metadata fields are arranged to achieve:

- a) Sortability - Near the beginning of the filename resides primary time oriented fields such as Sol and Spacecraft Clock Start Count (SCLK). This allows for sorting of files on the file system by spacecraft data acquisition time as events occurred on Mars.
- b) Readability - An effort is made to alternate Integer fields with ASCII character fields to optimize differentiation of field boundaries for the human user.

Please refer to the table in Appendix F for the list of spacecraft clock time (SCLK) recorded in the filename of data products.

## 18.1 EDR/RDR Single-Frame Filename

The M2020 camera data single-frame product filename convention is illustrated in Table 18-1 below.

**Table 18-1 – Camera instrument single-frame filenames convention**

Single-Frame Filename Convention

Field	Position (size, type)	Description
Instrument	01 (2, a)	<p>Instrument Identifier:</p> <ul style="list-style-type: none"> <li>FL : Front Hazcam Left (RCE-A)</li> <li>FR : Front Hazcam Right (RCE-A)</li> <li>FA : Front Hazcam Anaglyph (RCE-A)</li> <li>FG : Front Hazcam Colorglyph (RCE-A)</li> <li>BL : Front Hazcam Left (RCE-B)</li> <li>BR : Front Hazcam Right (RCE-B)</li> <li>BA : Front Hazcam Anaglyph (RCE-B)</li> <li>BG : Front Hazcam Colorglyph (RCE-B)</li> <li>CC : Cache Cam</li> <li>EA : EDL Parachute Uplook Cam A (PUC-A)</li> <li>EB : EDL Parachute Uplook Cam B (PUC-B)</li> <li>EC : EDL Parachute Uplook Cam C (PUC-C)</li> <li>ED : EDL Rover Downlook Cam (RDC)</li> <li>EL : EDL Lander Vision System (LCAM)</li> <li>EM: EDL Microphone</li> <li>ES : EDL Descent Stage Downlook Cam (DSD)</li> <li>EU : EDL Rover Uplook Cam (RUC)</li> <li>HN : Helicopter Navigation Cam</li> <li>HS : Helicopter Return To Earth Cam</li> <li>HA: Helicopter Nav Anaglyph</li> <li>HG: Helicopter RTE Colorglyph</li> <li>ZL : Mastcam-Z Left</li> <li>ZR : Mastcam-Z Right</li> <li>ZA : Mastcam-Z Anaglyph</li> <li>ZG : Mastcam-Z Colorglyph</li> <li>WS : MEDA Skycam</li> <li>NL : Navcam Left</li> <li>NR : Navcam Right</li> <li>NA : Navcam Anaglyph</li> <li>NG : Navcam Colorglyph</li> <li>PC : PIXL Micro Context Cam (MCC)</li> </ul>

## Single-Frame Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																															

Field	Position (size, type)	Description																				
		<ul style="list-style-type: none"><li>• RL : Rear Hazcam Left</li><li>• RR : Rear Hazcam Right</li><li>• RA : Rear Hazcam Anaglyph</li><li>• RG : Rear Hazcam Colorglyph</li><li>• SC : SHERLOC Context Imager (ACI)</li><li>• SE : SHERLOC Engineering (Imagers)</li><li>• SI : SHERLOC Imaging (Watson)</li><li>• SA : SHERLOC Imaging (Watson/ACI) Anaglyph</li><li>• SG : SHERLOC Imaging (Watson/ACI) Colorglyph</li><li>• SL : Watson/ACI treated as left-eye</li><li>• SR : Watson/ACI treated as right-eye</li><li>• LR : Supercam RMI</li></ul> <p>Non-imaging instrument identifiers (not described in this SIS):</p> <ul style="list-style-type: none"><li>• WE : MEDA Environment</li><li>• OX : MOXIE</li><li>• PE : PIXL Engineering</li><li>• PS : PIXL Spectrometer</li><li>• LS : SuperCam Non-Imaging Data</li><li>• SS : SHERLOC Spectrometer</li><li>• XM : RIMFAX Mobile</li><li>• XS : RIMFAX Stationary</li></ul>																				
Color/Filter	03 (1, i/a)	<p>Color flag (see Section X.X):</p> <ul style="list-style-type: none"><li>• E : Raw Bayer pattern</li><li>• M : Grayscale image (Monochrome/Panchromatic)</li><li>• A : Upper green bayer cells (G1 see Section X.X)</li><li>• D : Lower green bayer cells (G2 see Section X.X)</li><li>• O : Other</li><li>• _ : N/A</li></ul> <p>For all color cameras, the flag is set based on the color space depending on whether the product is a 3-band product, or an individual band.</p> <table><tr><th>Color Type</th><th>3-Band</th><th>Band 1</th><th>Band 2</th><th>Band 3</th></tr><tr><td>RGB</td><td>F</td><td>R</td><td>G</td><td>B</td></tr><tr><td>XYZ</td><td>T</td><td>X</td><td>Y</td><td>Z</td></tr><tr><td>xyY</td><td>C</td><td>J</td><td>K</td><td>L</td></tr></table>	Color Type	3-Band	Band 1	Band 2	Band 3	RGB	F	R	G	B	XYZ	T	X	Y	Z	xyY	C	J	K	L
Color Type	3-Band	Band 1	Band 2	Band 3																		
RGB	F	R	G	B																		
XYZ	T	X	Y	Z																		
xyY	C	J	K	L																		

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					-	TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																														

Field	Position (size, type)	Description																											
		<div><div>HSI</div><div>P</div><div>H</div><div>S</div><div>I</div></div> <p>For certain instruments, this flag may take on additional values indicating illumination or filter:</p> <p>For PIXL, the flag can be set based on the LEDs used to illuminate the target.</p> <table><thead><tr><th>LED Color</th><th>Flag</th></tr></thead><tbody><tr><td>Red</td><td>R</td></tr><tr><td>Green</td><td>G</td></tr><tr><td>Blue</td><td>B</td></tr><tr><td>Multiple</td><td>W</td></tr><tr><td>UV</td><td>U</td></tr><tr><td>SLI-A (Dense)</td><td>D</td></tr><tr><td>SLI-B (Sparse)</td><td>S</td></tr><tr><td>Off</td><td>—</td></tr></tbody></table> <p>For ZCAM and SHERLOC Imaging the flag can be filter or cover state.</p> <table><thead><tr><th>Instrument</th><th>Filters/Camera State</th><th>Description</th></tr></thead><tbody><tr><td>Mastcam-Z</td><td>0 – 7</td><td>See section X on ZCAM filters</td></tr><tr><td>Sherloc</td><td>0 - 3</td><td>Cover and LED state:  0 = cover closed, LEDs off 1 = cover open, LEDs off 2 = cover closed, LEDs on 3 = cover open, LEDs on</td></tr></tbody></table> <p>Other color flags may be defined in the future.</p> <p><b>Only “E”, “F”, or “M” or Filter/LED can appear in EDRs.</b></p>	LED Color	Flag	Red	R	Green	G	Blue	B	Multiple	W	UV	U	SLI-A (Dense)	D	SLI-B (Sparse)	S	Off	—	Instrument	Filters/Camera State	Description	Mastcam-Z	0 – 7	See section X on ZCAM filters	Sherloc	0 - 3	Cover and LED state:  0 = cover closed, LEDs off 1 = cover open, LEDs off 2 = cover closed, LEDs on 3 = cover open, LEDs on
	LED Color	Flag																											
	Red	R																											
	Green	G																											
	Blue	B																											
	Multiple	W																											
	UV	U																											
	SLI-A (Dense)	D																											
	SLI-B (Sparse)	S																											
	Off	—																											
Instrument	Filters/Camera State	Description																											
Mastcam-Z	0 – 7	See section X on ZCAM filters																											
Sherloc	0 - 3	Cover and LED state:  0 = cover closed, LEDs off 1 = cover open, LEDs off 2 = cover closed, LEDs on 3 = cover open, LEDs on																											

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					-	TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																														

Field	Position (size, type)	Description
		<b>For Bayered images, the filter is replaced by the color flag at the debayering step.</b>
Special flag	04 (1, a)	<p>Special Processing flag, applicable to RDRs only. EDRs always have "_".</p> <p>The special processing character is used to indicate off-nominal or special processing of the image. Examples include use of different correlation parameters, special stretches to eliminate shadows, reprocessing with different camera pointing, etc.</p> <p>The meaning of any individual character in this field (other than "_" which means nominal processing) will be defined on an ad-hoc basis as needed during the mission. Within one Sol or a range of sols, the character will be used consistently. So, this field can be used to group together all derived products resulting from one kind of special processing. An attempt will be made to maintain consistency across different sols as well, but this may not always be possible; thus the meaning of characters may change across different individual or ranges of sols.</p> <p>A database will be maintained containing all special processing designators that are used, the sols they relate to, and a description of the special processing that was done. This information is included in Appendix G.</p>
Primary timestamp	05 (4, i/a)	<p>Primary Timestamp that is of coarser granularity than the Secondary timestamp (documented later). Value type is based on either of four scenarios:</p> <p><u>Flight Cruise</u>  Year-DOY (4 alphanumeric) - This field stores two metadata items in the order:  a) One alpha character in range "A-Z" to designate Earth Year portion of the UTC-like time value, representing Years 2017 to 2042  b) Three integers in range "001-365" representing Day-of-Year (DOY)</p> <p><u>Flight Surface</u>  Sol (4 integer) - This field stores the 4-integer Sol (Mars solar day) of the <u>first</u> (i.e., lowest Clock time) acquired instrument data.</p> <p><u>Ground Test in which SCLK in NOT reset</u>  When SCLK continuously increments and does NOT repeat, there are two variants:  a) Year-DOY (4 alphanumeric) - This field stores two metadata items in the order:  1. One alpha character in range "A-Z" to designate Earth Year portion of the UTC-like time value, representing Years 2017 to 2042  2. Three integers in range "001-365" representing Day-of-Year (DOY)  – OR –  b) Sol (4 integer) - This field stores the 4-integer Sol (Mars solar day) of the <u>first</u> (i.e., lowest Clock time) acquired instrument data.</p>

## Single-Frame Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAI	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	EXT																																

Field	Position (size, type)	Description																																									
		<u>Ground Test in which SCLK is reset</u> When SCLK is reset and repeats, we lose time “uniqueness”. So, we have to change from SCLK to using “wall clock” derived from ERT and represent with a UTC-like format:  DOY-Year (4 alphanumeric) - This field stores two metadata items in reverse order compared to the previous “Year-DOY” cases, indicating that the Secondary Time field(described later) contains ERT a) Three integers in range “001-365” representing Day-of-Year (DOY) One alpha character in range “A-Z” to designate Earth Year portion of the UTC-like time value, representing Years 2017 to 2042 This field’s value type (Sol or Year/DOY) dictates the value type for the Secondary time field (SCLK or UTC-like).																																									
		<table><tr><th>Scenario</th><th>Time Type</th><th>Value Format</th><th>Valid Values</th><th>Time Range</th></tr><tr><td rowspan="4">Flight Cruise</td><td rowspan="4">Year-DOY</td><td>A&lt;ddd&gt;, B&lt;ddd&gt;, . . . Z&lt;ddd&gt;</td><td>A001, A002, ... A365, B001, B002, ... B365, . . . Z001, Z002, ..., Z365</td><td>2017 DOY 1 – 365, 2018 DOY 1 – 365, . . . 2042 DOY 1 – 365</td></tr><tr><td>&lt;aaaa&gt;</td><td>“_____” (4 underscores)</td><td>Value is out of range</td></tr><tr><td rowspan="2">Flight Surface</td><td rowspan="2">Sol</td><td>&lt;nnnn&gt;</td><td>0000, 0001, ... 9999</td><td>0 thru 9999</td></tr><tr><td>&lt;aaaa&gt;</td><td>“_____” (4 underscores)</td><td>Value is out of range</td></tr><tr><td rowspan="3">Ground Test SCLK NOT reset</td><td rowspan="2">Year-DOY</td><td>(same as Flight Cruise)</td><td>(same as Flight Cruise)</td><td>(same as Flight Cruise)</td></tr><tr><td rowspan="2">Sol</td><td>&lt;nnnn&gt;</td><td>“0000”, “0001”, ... “9999”</td><td>0 thru 9999</td></tr><tr><td>&lt;aaaa&gt;</td><td>“_____” (4 underscores)</td><td>Value is out of range</td></tr><tr><td rowspan="3">Ground Test SCLK Reset</td><td rowspan="3">DOY-Year</td><td>&lt;ddd&gt;A, &lt;ddd&gt;B, . . . &lt;ddd&gt;Z</td><td>001A, 002A, ... 365A, 001B, 002B, ... 365B, . . . 001Z, 002Z, ... 365Z</td><td>DOY 1 – 365 2017, DOY 1 – 365 2018, . . . DOY 1 – 365 2042</td></tr><tr><td>&lt;aaaa&gt;</td><td>“_____” (4 underscores)</td><td>Value is out of range</td></tr></table>	Scenario	Time Type	Value Format	Valid Values	Time Range	Flight Cruise	Year-DOY	A<ddd>, B<ddd>, . . . Z<ddd>	A001, A002, ... A365, B001, B002, ... B365, . . . Z001, Z002, ..., Z365	2017 DOY 1 – 365, 2018 DOY 1 – 365, . . . 2042 DOY 1 – 365	<aaaa>	“_____” (4 underscores)	Value is out of range	Flight Surface	Sol	<nnnn>	0000, 0001, ... 9999	0 thru 9999	<aaaa>	“_____” (4 underscores)	Value is out of range	Ground Test SCLK NOT reset	Year-DOY	(same as Flight Cruise)	(same as Flight Cruise)	(same as Flight Cruise)	Sol	<nnnn>	“0000”, “0001”, ... “9999”	0 thru 9999	<aaaa>	“_____” (4 underscores)	Value is out of range	Ground Test SCLK Reset	DOY-Year	<ddd>A, <ddd>B, . . . <ddd>Z	001A, 002A, ... 365A, 001B, 002B, ... 365B, . . . 001Z, 002Z, ... 365Z	DOY 1 – 365 2017, DOY 1 – 365 2018, . . . DOY 1 – 365 2042	<aaaa>	“_____” (4 underscores)	Value is out of range
		Scenario	Time Type	Value Format	Valid Values	Time Range																																					
		Flight Cruise	Year-DOY	A<ddd>, B<ddd>, . . . Z<ddd>	A001, A002, ... A365, B001, B002, ... B365, . . . Z001, Z002, ..., Z365	2017 DOY 1 – 365, 2018 DOY 1 – 365, . . . 2042 DOY 1 – 365																																					
				<aaaa>	“_____” (4 underscores)	Value is out of range																																					
				Flight Surface	Sol	<nnnn>	0000, 0001, ... 9999	0 thru 9999																																			
						<aaaa>	“_____” (4 underscores)	Value is out of range																																			
		Ground Test SCLK NOT reset	Year-DOY	(same as Flight Cruise)	(same as Flight Cruise)	(same as Flight Cruise)																																					
				Sol	<nnnn>	“0000”, “0001”, ... “9999”	0 thru 9999																																				
			<aaaa>		“_____” (4 underscores)	Value is out of range																																					
Ground Test SCLK Reset	DOY-Year	<ddd>A, <ddd>B, . . . <ddd>Z	001A, 002A, ... 365A, 001B, 002B, ... 365B, . . . 001Z, 002Z, ... 365Z	DOY 1 – 365 2017, DOY 1 – 365 2018, . . . DOY 1 – 365 2042																																							
		<aaaa>	“_____” (4 underscores)	Value is out of range																																							
		Venue	9 (1, a)	Mission venue identifier: <ul style="list-style-type: none"><li>_ : Flight (surface or cruise)</li><li>A : AVSTB</li></ul>																																							

## Single-Frame Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																															

Field	Position (size, type)	Description																							
		<ul style="list-style-type: none"><li>F : FSWTB</li><li>M : MSTB</li><li>R : "ROASTT"</li><li>S : "Scarecrow"</li><li>V : VSTB</li></ul> <p>Other venue identifiers may be defined later.</p>																							
Secondary timestamp	10 (10, i)	<p>Secondary timestamp that is of finer granularity than the Primary timestamp. Value type is based on either of four scenarios:</p> <p><u>Flight Cruise</u> SCLK – This field stores the 10-integer SCLK (seconds). Which specific SCLK count (Start or End) is used depends on the instrument, but nominally it is the starting count of the <u>first</u> (i.e., lowest Clock time) acquired instrument data.</p> <p><u>Flight Surface</u> SCLK – Same as for "Flight Cruise"</p> <p><u>Ground Test in which SCLK in NOT reset</u> SCLK – Same as for "Flight Cruise"</p> <p><u>Ground Test in which SCLK is reset</u> ERT - This field stores the ERT time portions Month, Day-of-month, Hour and Seconds as 10 integers in a UTC-like format</p> <p>This field's value type associates with the value type of the Primary timestamp field.</p> <table><tr><th>Scenario</th><th>Time Type</th><th>Value Format</th><th>Valid Values</th><th>Time Range</th></tr><tr><td rowspan="2">Flight Cruise</td><td rowspan="2">SCLK</td><td>&lt;ssssssssss&gt; (Seconds)</td><td>0000000000, 0000000001, • • • 9999999999</td><td>0 thru 9999999999</td></tr><tr><td>&lt;aaaaaaaa&gt;</td><td>"-----" (10 underscores)</td><td>Value is out of range</td></tr><tr><td>Flight Surface</td><td>SCLK</td><td>(same as Flight Cruise)</td><td>(same as Flight Cruise)</td><td>(same as Flight Cruise)</td></tr><tr><td>Ground Test SCLK NOT reset</td><td>SCLK</td><td>(same as Flight Cruise)</td><td>(same as Flight Cruise)</td><td>(same as Flight Cruise)</td></tr></table>	Scenario	Time Type	Value Format	Valid Values	Time Range	Flight Cruise	SCLK	<ssssssssss> (Seconds)	0000000000, 0000000001, • • • 9999999999	0 thru 9999999999	<aaaaaaaa>	"-----" (10 underscores)	Value is out of range	Flight Surface	SCLK	(same as Flight Cruise)	(same as Flight Cruise)	(same as Flight Cruise)	Ground Test SCLK NOT reset	SCLK	(same as Flight Cruise)	(same as Flight Cruise)	(same as Flight Cruise)
Scenario	Time Type	Value Format	Valid Values	Time Range																					
Flight Cruise	SCLK	<ssssssssss> (Seconds)	0000000000, 0000000001, • • • 9999999999	0 thru 9999999999																					
		<aaaaaaaa>	"-----" (10 underscores)	Value is out of range																					
Flight Surface	SCLK	(same as Flight Cruise)	(same as Flight Cruise)	(same as Flight Cruise)																					
Ground Test SCLK NOT reset	SCLK	(same as Flight Cruise)	(same as Flight Cruise)	(same as Flight Cruise)																					

This document has been reviewed and determined not to contain export controlled technical data.

## Single-Frame Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																															

Field	Position (size, type)	Description								
		<table><tr><td rowspan="2">Ground Test SCLK reset</td><td rowspan="2">ERT</td><td>&lt;MMDDHHmmss&gt; (Month, Day-of- month, Hour, Minute, Second)</td><td>“0101010000”, “0101010001”, • • • “1231235959”</td><td>January 1, 01:00:00 thru December 31, 23:59:59</td></tr><tr><td>&lt;aaaaaaaa&gt;</td><td>“-----” (10 underscores)</td><td>Value is out of range</td></tr></table>	Ground Test SCLK reset	ERT	<MMDDHHmmss> (Month, Day-of- month, Hour, Minute, Second)	“0101010000”, “0101010001”, • • • “1231235959”	January 1, 01:00:00 thru December 31, 23:59:59	<aaaaaaaa>	“-----” (10 underscores)	Value is out of range
		Ground Test SCLK reset			ERT	<MMDDHHmmss> (Month, Day-of- month, Hour, Minute, Second)	“0101010000”, “0101010001”, • • • “1231235959”	January 1, 01:00:00 thru December 31, 23:59:59		
<aaaaaaaa>	“-----” (10 underscores)		Value is out of range							
		<p><u>ZCAM/SHERLOC Video Sequence</u></p> <p>Secondary Timestamp + Milliseconds for ith frame is computed as follows: First_Frame_SCLK + framerate * i</p> <p>Frame rate calculation (as in MSSS M2020 Camera Common Command): frametime = 1/sysclk*(400*(2*vflush+ul_y+lines)+(lines*1648*4))+exposure framerate = 1/(frametime+interval)</p> <p>where sysclk is 40e6 for mode 0, 20e6 for mode 9, and 10e6 for mode 1 mode is the camera head mode used to acquire the image vflush is the value of the low- order 12 bits of parameter 2 times 2 ul_y is the upper left Y coordinate of the image (specified in bits) lines is the number of lines in the output image (specified in bits) exposure is the exposure time in seconds 1648 is total number of samples per line interval: the time delay between images in units of 1 millisecond. If data acquired at the fastest possible frame rate, the value is 0. (This equation only applies if the video exposure mode is not being used. For 1280x720 video, video exposure mode increases the frame rate by about 1.75x.)"</p>								
_/_Mesh code	20 (1, a)	Underscore for readability. Always “_” for most files. For single-frame meshes using this convention, this is set to a code indicating the type of geometry used for the mesh.  M = masked XYZ X = unmasked XYZ								
Milliseconds	21 (3, i)	Milliseconds of either the SCLK or UTC.								
Product type	24 (3, a)	Product identifier.  EDR: See Table 5-3 for details RDR: See Table 17-2 and Table 17-3 for details.								
Geometry	27 (1, a)	Geometry flag: <ul style="list-style-type: none"><li>_ : Non-linearized (raw geometry)</li><li>L : Product has been linearized with actual stereo partner. Check the</li></ul>								

This document has been reviewed and determined not to contain export controlled technical data.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					-	TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																														

Field	Position (size, type)	Description																																						
		<p>stereo processing counter to see if it uses the nominal partner or a different partner.</p> <ul style="list-style-type: none"><li>T : PIXL MCC Trapezoidal correction has been applied.</li></ul> <p>Note that on Mars2020 there is no "nominal" linearization as on other missions. All linearization is actual, so we use "L" as the flag.</p> <p>All EDRs are Raw geometry (“_”).</p>																																						
Thumbnail	28 (1, a)	<p>Thumbnail flag:</p> <ul style="list-style-type: none"><li>T : Product is a thumbnail</li><li>N : Nominal Product is a non-thumbnail (full-frame, sub-frame, downsample)</li></ul>																																						
Site/Flight/Set	29 (3, i/a)	<p>Site/Flight/Set identifier; first element of the RMC (see Section 4.4).</p> <p>Site location count from the RMC where the data was acquired for the rover. Flight number from the RMC where the data was acquired for the helicopter. Set number from the RMC where the data was acquired for LCAM (always 0).</p> <table><thead><tr><th>Values</th><th>Range</th></tr></thead><tbody><tr><td>000, 001, ..., 999</td><td>0 thru 999</td></tr><tr><td>A00, A01, ..., A99</td><td>1000 thru 1099</td></tr><tr><td>B00, B01, ..., B99</td><td>1100 thru 1199</td></tr><tr><td>...</td><td>...</td></tr><tr><td>Z00, Z01, ... Z99</td><td>3500 thru 3599</td></tr><tr><td>AA0, AA1, ..., AA9</td><td>3600 thru 3609</td></tr><tr><td>AB0, AB1, ..., AB9</td><td>3610 thru 3619</td></tr><tr><td>...</td><td>...</td></tr><tr><td>ZZ0, ZZ1, ..., ZZ9</td><td>10350 thru 10359</td></tr><tr><td>AAA, AAB, ..., AAZ</td><td>10360 thru 10385</td></tr><tr><td>ABA, ABB, ..., ABZ</td><td>10386 thru 10411</td></tr><tr><td>...</td><td>...</td></tr><tr><td>ZZA, ZZB, ..., ZZZ</td><td>27910 thru 27935</td></tr><tr><td>0AA, 0AB, ..., 0AZ</td><td>27936 thru 27961</td></tr><tr><td>0BA, 0BB, ..., 0BZ</td><td>27962 thru 27987</td></tr><tr><td>...</td><td>...</td></tr><tr><td>7CA, 7CB, ..., 7CZ</td><td>32720 thru 32745</td></tr><tr><td>7DA, 7DB, ..., 7DV</td><td>32746 thru 32767</td></tr></tbody></table>	Values	Range	000, 001, ..., 999	0 thru 999	A00, A01, ..., A99	1000 thru 1099	B00, B01, ..., B99	1100 thru 1199	...	...	Z00, Z01, ... Z99	3500 thru 3599	AA0, AA1, ..., AA9	3600 thru 3609	AB0, AB1, ..., AB9	3610 thru 3619	...	...	ZZ0, ZZ1, ..., ZZ9	10350 thru 10359	AAA, AAB, ..., AAZ	10360 thru 10385	ABA, ABB, ..., ABZ	10386 thru 10411	...	...	ZZA, ZZB, ..., ZZZ	27910 thru 27935	0AA, 0AB, ..., 0AZ	27936 thru 27961	0BA, 0BB, ..., 0BZ	27962 thru 27987	...	...	7CA, 7CB, ..., 7CZ	32720 thru 32745	7DA, 7DB, ..., 7DV	32746 thru 32767
Values	Range																																							
000, 001, ..., 999	0 thru 999																																							
A00, A01, ..., A99	1000 thru 1099																																							
B00, B01, ..., B99	1100 thru 1199																																							
...	...																																							
Z00, Z01, ... Z99	3500 thru 3599																																							
AA0, AA1, ..., AA9	3600 thru 3609																																							
AB0, AB1, ..., AB9	3610 thru 3619																																							
...	...																																							
ZZ0, ZZ1, ..., ZZ9	10350 thru 10359																																							
AAA, AAB, ..., AAZ	10360 thru 10385																																							
ABA, ABB, ..., ABZ	10386 thru 10411																																							
...	...																																							
ZZA, ZZB, ..., ZZZ	27910 thru 27935																																							
0AA, 0AB, ..., 0AZ	27936 thru 27961																																							
0BA, 0BB, ..., 0BZ	27962 thru 27987																																							
...	...																																							
7CA, 7CB, ..., 7CZ	32720 thru 32745																																							
7DA, 7DB, ..., 7DV	32746 thru 32767																																							

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					-	TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																														

Field	Position (size, type)	Description																														
		Value out of range																														
Drive/Pos/ Instance	32 (4, i/a)	<p>Drive/Pos/Instance identifier; second element of the RMC (see Section 4.4).</p> <p>Drive count (position within a Site location) from the RMC where the data was acquired for the rover.</p> <p>Pos count from the RMC where the data was acquired for the helicopter (always 1 due to bug, see Section 4.4.2.2).</p> <p>Instance number from the RMC where the data was acquired for LCAM.</p> <table><thead><tr><th>Values</th><th>Range</th></tr></thead><tbody><tr><td>0000, 0001, ..., 9999</td><td>0 thru 9999</td></tr><tr><td>A000, A001, ..., A999</td><td>10000 thru 10999</td></tr><tr><td>B000, B001, ..., B999</td><td>11000 thru 11999</td></tr><tr><td>...</td><td>...</td></tr><tr><td>Z000, Z001, ... Z999</td><td>35000 thru 35999</td></tr><tr><td>AA00, AA01, ..., AA99</td><td>36000 thru 36099</td></tr><tr><td>AB00, AB01, ..., AB99</td><td>36100 thru 36199</td></tr><tr><td>...</td><td>...</td></tr><tr><td>AZ00, AZ01, ..., AZ99</td><td>38500 thru 38599</td></tr><tr><td>BA00, BA01, ..., BA99</td><td>38600 thru 38699</td></tr><tr><td>BB00, BB01, ..., BB99</td><td>38700 thru 38799</td></tr><tr><td>...</td><td>...</td></tr><tr><td>LJ00, LJ01, ..., LJ35</td><td>65500 thru 65535</td></tr><tr><td>Value is out of range</td><td></td></tr></tbody></table>	Values	Range	0000, 0001, ..., 9999	0 thru 9999	A000, A001, ..., A999	10000 thru 10999	B000, B001, ..., B999	11000 thru 11999	...	...	Z000, Z001, ... Z999	35000 thru 35999	AA00, AA01, ..., AA99	36000 thru 36099	AB00, AB01, ..., AB99	36100 thru 36199	...	...	AZ00, AZ01, ..., AZ99	38500 thru 38599	BA00, BA01, ..., BA99	38600 thru 38699	BB00, BB01, ..., BB99	38700 thru 38799	...	...	LJ00, LJ01, ..., LJ35	65500 thru 65535	Value is out of range	
Values	Range																															
0000, 0001, ..., 9999	0 thru 9999																															
A000, A001, ..., A999	10000 thru 10999																															
B000, B001, ..., B999	11000 thru 11999																															
...	...																															
Z000, Z001, ... Z999	35000 thru 35999																															
AA00, AA01, ..., AA99	36000 thru 36099																															
AB00, AB01, ..., AB99	36100 thru 36199																															
...	...																															
AZ00, AZ01, ..., AZ99	38500 thru 38599																															
BA00, BA01, ..., BA99	38600 thru 38699																															
BB00, BB01, ..., BB99	38700 thru 38799																															
...	...																															
LJ00, LJ01, ..., LJ35	65500 thru 65535																															
Value is out of range																																
Seq-ID/RTT	36 (9, i/a)	<p>Identifies sequence-ID or Round-Trip Tracking token (RTT).</p> <ul style="list-style-type: none"><li>Sequence-ID : Identifier indicating the command sequenced the image was acquired from. Specific values will be assigned by the uplink team.</li><li>RTT : Unique identifier used for tracking activities. Used in filename for PIXL ONLY.</li></ul> <p>Note: All cameras will have an RTT associated with their images but not included in the filename. PIXL is a special case where RTT and PMC are required for filename uniqueness.</p>																														
Camera specific	45 (4, i/a)	<p>Camera specific identifier.</p> <p>Each instrument has a unique format:</p>																														

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					-	TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																														

Field	Position (size, type)	Description																								
		<table><tr><th>Instrument</th><th>Format</th></tr><tr><td>ECAM Tile</td><td>STT_</td></tr><tr><td>ECAM Reconstructed</td><td>S0DR</td></tr><tr><td>ZCAM</td><td>SZZZ</td></tr><tr><td>SCAM</td><td>SNNN</td></tr><tr><td>PIXL MCC</td><td>PPPP</td></tr><tr><td>SHERLOC</td><td>SENN</td></tr><tr><td>MEDA Skycam</td><td>S000</td></tr><tr><td>EDL Cameras</td><td>S000</td></tr><tr><td>EDL Microphone</td><td>_III</td></tr><tr><td>EDL Video</td><td>_VVV</td></tr><tr><td>HELI</td><td>S000</td></tr></table>	Instrument	Format	ECAM Tile	STT_	ECAM Reconstructed	S0DR	ZCAM	SZZZ	SCAM	SNNN	PIXL MCC	PPPP	SHERLOC	SENN	MEDA Skycam	S000	EDL Cameras	S000	EDL Microphone	_III	EDL Video	_VVV	HELI	S000
		Instrument	Format																							
		ECAM Tile	STT_																							
		ECAM Reconstructed	S0DR																							
		ZCAM	SZZZ																							
		SCAM	SNNN																							
		PIXL MCC	PPPP																							
		SHERLOC	SENN																							
		MEDA Skycam	S000																							
		EDL Cameras	S000																							
		EDL Microphone	_III																							
		EDL Video	_VVV																							
		HELI	S000																							
		Format key:																								
		<ul style="list-style-type: none"><li>D : Single-resolution or multi-resolution downsampling (ECAM Reconstructed images only). Multi-resolution value represents the highest resolution tile(s) of the reconstructed image.</li></ul>																								
		<table><tr><th>Valid values</th><th>Description</th></tr><tr><td>A</td><td>Single resolution (Alone)</td></tr><tr><td>M</td><td>Multi-resolution</td></tr></table>	Valid values	Description	A	Single resolution (Alone)	M	Multi-resolution																		
		Valid values	Description																							
		A	Single resolution (Alone)																							
		M	Multi-resolution																							
		<ul style="list-style-type: none"><li>NNN : Point Number (RMI only). SuperCam MU point number that is reported in the ScamAncillaryArgs DPO. This value indicates the point number of a raster activity. Valid values are 000-099.</li><li>ENN : E represents the experiment ID and NN represents the total number of images that were collected as part of the experiment. Valid values are 0-9 for E and 0-99 for NN.</li><li>PPPP : PIXL motion counter, as a hexadecimal number. S: Stereo partner counter.</li><li>R : Reconstruction counter (ECAM Reconstructed only). When a new tile is added that changes either the geometry or color of the reconstructed image, the counter increments by one. Valid values are 1-9, then A-Z.</li><li>S : Stereo partner counter.</li></ul>																								

This document has been reviewed and determined not to contain export controlled technical data.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					-	TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																														

Field	Position (size, type)	Description												
		<p>All EDRs are “_”.</p> <table><tr><th>Valid values</th><th>Description</th></tr><tr><td>—</td><td>Nominal (synchronized) stereo or No nominal stereo partner</td></tr><tr><td>A-Z then 0-9</td><td>Indicates off-nominal stereo partern ( see 8.1)</td></tr></table>	Valid values	Description	—	Nominal (synchronized) stereo or No nominal stereo partner	A-Z then 0-9	Indicates off-nominal stereo partern ( see 8.1)						
		Valid values	Description											
—	Nominal (synchronized) stereo or No nominal stereo partner													
A-Z then 0-9	Indicates off-nominal stereo partern ( see 8.1)													
		<ul style="list-style-type: none"><li>• TT: Tile ID(ECAM Tile only). Identifies a unique tile within a reconstructed image. Valid values are 00-99.</li><li>• ZZZ :Focal length in millimeters (ZCAM only).Valid values are 000-999. Commanded value is used, unless it’s cruise fsw where it’s unavailable and actual value is used instead.</li><li>• III : the index number of the audio clip</li><li>• VVV : Frame number of the video</li><li>• _ : Literal underscore. As last character of camera specific field, identifies an ECAM Tile.</li><li>• 0 : Literal zero. Undefined field placeholder.</li></ul> <p>Instrument teams with undefined (“0”) fields may define values at a later date.</p>												
Downsample	49 (1, i/a)	<p>Downsample resolution identifier.</p> <p>This value (n) indicates the level of downsampling applied to the image by the following equation:</p> <p>Resolution = 2<sup>n</sup> x 2<sup>n</sup></p> <table><tr><th>Valid values</th><th>Resolution</th></tr><tr><td>0</td><td>1x1</td></tr><tr><td>1</td><td>2x2</td></tr><tr><td>2</td><td>4x4</td></tr><tr><td>3</td><td>8x8</td></tr><tr><td>...</td><td>...</td></tr></table> <p>For the case of reconstructed ECAM images, this value represents the highest resolution tile(s) in the reconstructed image (least amount of downsampling).</p>	Valid values	Resolution	0	1x1	1	2x2	2	4x4	3	8x8	...	...
Valid values	Resolution													
0	1x1													
1	2x2													
2	4x4													
3	8x8													
...	...													
Compression	50 (2, i/a)	Compression type identifier.												

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					-	TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																														

Field	Position (size, type)	Description																					
		<p>There are several modes of compression available, varying per instrument. The uplink team will decide which algorithm will provide the best results on a per sequence basis.</p> <table><tr><th>Type</th><th>Valid values</th><th>Description</th></tr><tr><td>JPEG (lossy)</td><td>00 01-99 A0</td><td>Thumbnail, or JPEG at unknown quality Jpeg quality level Jpeg quality level 100</td></tr><tr><td>ICER (lossy)</td><td>I1, I2, ..., I8 I9</td><td>1 bpp, 2 bpp, ..., 8 bpp Anything higher than 8 bpp</td></tr><tr><td rowspan="4">Lossless</td><td>LI</td><td>ICER</td></tr><tr><td>LL</td><td>LOCO</td></tr><tr><td>LM</td><td>MSSS</td></tr><tr><td>LU</td><td>Uncompressed</td></tr><tr><td>Other</td><td>FH</td><td>Frame extracted from H.264 video</td></tr></table> <p>For ECAM reconstructed images, compression value represents the compression level of the best tile(s) in the reconstruction.</p>	Type	Valid values	Description	JPEG (lossy)	00 01-99 A0	Thumbnail, or JPEG at unknown quality Jpeg quality level Jpeg quality level 100	ICER (lossy)	I1, I2, ..., I8 I9	1 bpp, 2 bpp, ..., 8 bpp Anything higher than 8 bpp	Lossless	LI	ICER	LL	LOCO	LM	MSSS	LU	Uncompressed	Other	FH	Frame extracted from H.264 video
		Type	Valid values	Description																			
		JPEG (lossy)	00 01-99 A0	Thumbnail, or JPEG at unknown quality Jpeg quality level Jpeg quality level 100																			
		ICER (lossy)	I1, I2, ..., I8 I9	1 bpp, 2 bpp, ..., 8 bpp Anything higher than 8 bpp																			
		Lossless	LI	ICER																			
			LL	LOCO																			
			LM	MSSS																			
			LU	Uncompressed																			
		Other	FH	Frame extracted from H.264 video																			
		Producer	52 (1, a)	<p>Identifier for the institution/team that created this product:</p> <ul style="list-style-type: none"><li>J : JPL (IDS/MIPL)</li><li>A: ASU (Mastcam-Z team)</li><li>P : Principal investigator of instrument.</li></ul> <table><tr><th>Instrument</th><th>PI</th></tr><tr><td>ECAM</td><td>JPL</td></tr><tr><td>ZCAM</td><td>ASU (Tempe, AZ)</td></tr><tr><td>SCAM RMI</td><td>IRAP (France)</td></tr><tr><td>PIXL MCC</td><td>JPL</td></tr><tr><td>SHERLOC</td><td>JPL</td></tr><tr><td>MEDA Skycam</td><td>Ministry of Education and Science (Spain)</td></tr><tr><td>EDL Cameras</td><td>JPL</td></tr><tr><td>HELI NAV/RTE</td><td>JPL</td></tr></table>	Instrument	PI	ECAM	JPL	ZCAM	ASU (Tempe, AZ)	SCAM RMI	IRAP (France)	PIXL MCC	JPL	SHERLOC	JPL	MEDA Skycam	Ministry of Education and Science (Spain)	EDL Cameras	JPL	HELI NAV/RTE	JPL	
Instrument	PI																						
ECAM	JPL																						
ZCAM	ASU (Tempe, AZ)																						
SCAM RMI	IRAP (France)																						
PIXL MCC	JPL																						
SHERLOC	JPL																						
MEDA Skycam	Ministry of Education and Science (Spain)																						
EDL Cameras	JPL																						
HELI NAV/RTE	JPL																						

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					-	TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																														

Field	Position (size, type)	Description																				
		<ul style="list-style-type: none"><li>A – I, K – O, Q – Z : Co-I to be identified per instrument at the discretion of the instrument PI.</li><li>_ : undefined/other</li></ul> <p>Other producer codes will be added in the future.</p>																				
Version	53 (2, a)	<p>Product version number. Increments by one whenever a previously generated file with an otherwise identical filename exists.</p> <table><thead><tr><th>Values</th><th>Range</th></tr></thead><tbody><tr><td>01, 02 ..., 99</td><td>1 thru 99</td></tr><tr><td>A0, A1, ..., A9</td><td>100 thru 109</td></tr><tr><td>AA, AB, ..., AZ</td><td>110 thru 135</td></tr><tr><td>B0, B1, B2 ..., B9</td><td>136 thru 145</td></tr><tr><td>BA, BB, ..., BZ</td><td>146 thru 171</td></tr><tr><td>...</td><td>...</td></tr><tr><td>Z0, Z1, ..., Z9</td><td>1000 thru 1009</td></tr><tr><td>ZA, ZB, ..., ZZ</td><td>1010 thru 1035</td></tr><tr><td>--</td><td>Value is out of range</td></tr></tbody></table> <p>Every version need not exist. E.g. version 01, 02, and 04 may exist but not 03. In general, the highest-numbered version represents the best version of that product. This field increments independently of all fields.</p>	Values	Range	01, 02 ..., 99	1 thru 99	A0, A1, ..., A9	100 thru 109	AA, AB, ..., AZ	110 thru 135	B0, B1, B2 ..., B9	136 thru 145	BA, BB, ..., BZ	146 thru 171	...	...	Z0, Z1, ..., Z9	1000 thru 1009	ZA, ZB, ..., ZZ	1010 thru 1035	--	Value is out of range
Values	Range																					
01, 02 ..., 99	1 thru 99																					
A0, A1, ..., A9	100 thru 109																					
AA, AB, ..., AZ	110 thru 135																					
B0, B1, B2 ..., B9	136 thru 145																					
BA, BB, ..., BZ	146 thru 171																					
...	...																					
Z0, Z1, ..., Z9	1000 thru 1009																					
ZA, ZB, ..., ZZ	1010 thru 1035																					
--	Value is out of range																					
.	55 (1, a)	Separator for filename and extension. Always set to “.”																				
Extension	56 (3, a)	<p>File extension:</p> <ul style="list-style-type: none"><li>VIC : Used for VICAR files.</li><li>IMG : Same as .VIC with ODL label.</li><li>TIF : TIFF formatted image file. (no label)</li><li>JPG : JPEG formatted image file. (no label)</li><li>PNG : PNG formatted image file. (no label)</li><li>TXT : ASCII text file.</li><li>iv : Inventor-format file</li><li>ht : Height-map file (VICAR format)</li><li>rgb : Skin file in SGI RGB format</li><li>obj: Mesh file in Wavefront OBJ format</li><li>mtl : OBJ Material file</li></ul>																				

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT	COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP	VENUE	SECONDARY TIMESTAMP					-	TERTIARY TIMESTAMP	PROD TYPE	GEOMETRY THUMBNAIL	SITE	DRIVE	SEQUENCE/RTT					CAM SPECIFIC	DOWNSAMPLE	COMPRESSION	PRODUCER	VERSION	.	EXT																														

Field	Position (size, type)	Description
		<ul style="list-style-type: none"> <li>png : Browse and texture maps for meshes</li> <li>mlp : MeshLab project file</li> <li>xml : PDS label in xml format</li> <li>xmlf : filter file containing rover mask polygons in xml format</li> <li>WAV: WAV-format audio</li> <li>flac: Free Lossless Audio</li> </ul> <p>Note: other file extensions to be added in the future</p>

## 18.2 Mosaic RDR Filename

The filename convention used for M2020 mosaics is illustrated in Table 18-2 below.

**Table 18-2 – Camera instrument mosaic filenames convention**

Mosaic Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44					
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER	SPECIAL FLAG		SOL					MULTI SOL		PRODUCT TYPE			VENUE		SITE			DRIVE				MULTI DRIVE		PROJECTION			GEOM CORR		COORD FRAME		BRT CORR		DESCRIPTION					PRODUCER		VERSION				EXT	

Field	Position (size, type)	Description
Instrument 1	1 (1,a)	<p>First Instrument Identifier:</p> <ul style="list-style-type: none"> <li>N : Navcam</li> <li>Z : Mastcam-Z</li> <li>L : Supercam RMI</li> <li>P : PIXL Micro Context Cam (MCC)</li> <li>I : SHERLOC Imaging (WATSON)</li> <li>C : SHERLOC ACI</li> <li>F : Front Hazcam (RCE-A)</li> <li>B : Front Hazcam (RCE-B)</li> <li>R : Rear Hazcam</li> <li>E : EDL Camera</li> <li>V : Mars Helicopter Navigation Cam</li> <li>H : Mars Helicopter RTE Cam</li> </ul>

This document has been reviewed and determined not to contain export controlled technical data.

## Mosaic Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE		VENUE		SITE			DRIVE				MULTI DRIVE	PROJECTION		GEOM CORR	COORD FRAME	BRT CORR							PRODUCER	VERSION	.				EXT		

Field	Position (size, type)	Description															
Instrument 2	2 (1,a)	Second Instrument Identifier: <ul style="list-style-type: none"><li>N : Navcam</li><li>Z : Mastcam-Z</li><li>L : Supercam RMI</li><li>P : PIXL Micro Context Cam (MCC)</li><li>I : SHERLOC Imaging (WATSON)</li><li>C : SHERLOC ACI</li><li>F : Front Hazcam (RCE-A)</li><li>B : Front Hazcam (RCE-B)</li><li>R : Rear Hazcam</li><li>E : EDL Camera</li><li>V : Mars Helicopter Navigation Cam</li><li>X : Use when there are more than one instruments</li><li>_ : Use when there is only one instrument</li></ul>															
Eye	3 (1,a)	Identifies the eye of the data used to make this mosaic. For stereo mosaics, indicates which eye should view this product (in the unlikely event these two definitions are in conflict, the second use prevails for stereo mosaics). <ul style="list-style-type: none"><li>A : Anaglyph</li><li>G : Colorglyph</li><li>X : Mixed, used for by products which have more than one image.</li><li>L : Left eye</li><li>M : Mono</li><li>R : Right eye</li><li>S : Stereo 2-band</li></ul>															
Color/Filter	4 (3,a)	Indicates the filter or color used for each band of the mosaic. Single-band mosaics should repeat the same character 3 times. Three-band color spaces may either list the three one-band components (e.g. "RGB") or repeat the 3-band code 3 times (e.g. "FFF"). Valid values: <ul style="list-style-type: none"><li>0 or _ : Filter not applicable</li><li>M : Grayscale image (Monochrome/Panchromatic)</li><li>U : UV (PIXL only)</li></ul> <table><thead><tr><th>Color Type</th><th>3-Band</th><th>Band 1</th><th>Band 2</th><th>Band 3</th></tr></thead><tbody><tr><td>RGB</td><td>F</td><td>R</td><td>G</td><td>B</td></tr><tr><td>XYZ</td><td>T</td><td>X</td><td>Y</td><td>Z</td></tr></tbody></table>	Color Type	3-Band	Band 1	Band 2	Band 3	RGB	F	R	G	B	XYZ	T	X	Y	Z
Color Type	3-Band	Band 1	Band 2	Band 3													
RGB	F	R	G	B													
XYZ	T	X	Y	Z													

This document has been reviewed and determined not to contain export controlled technical data.



## Mosaic Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE		VENUE		SITE			DRIVE				MULTI DRIVE	PROJECTION		GEOM CORR	COORD FRAME	BRT CORR							DESCRIPTION	PRODUCER	VERSION				EXT		

Field	Position (size, type)	Description															
		<table><tr><td>xyY</td><td>C</td><td>J</td><td>K</td><td>L</td></tr><tr><td>HSI</td><td>P</td><td>H</td><td>S</td><td>I</td></tr></table> <p>Other color flags may be defined in the future.</p>	xyY	C	J	K	L	HSI	P	H	S	I					
xyY	C	J	K	L													
HSI	P	H	S	I													
Special	7 (1,a)	<p>Special Processing flag. The special processing character is used to indicate off-nominal or special processing of the image. Examples include use of different correlation parameters, special stretches to eliminate shadows, reprocessing with different camera pointing, etc.</p> <p>The meaning of any individual character in this field (other than "_" which means nominal processing) will be defined on an ad-hoc basis as needed during the mission. Within one Sol or a range of Sols, the character will be used consistently. So, this field can be used to group together all derived products resulting from one kind of special processing. An attempt will be made to maintain consistency across different Sols as well, but this may not always be possible; thus the meaning of characters may change across different individual or ranges of Sols.</p> <p>A database will be maintained containing all special processing designators that are used, the sols they relate to, and a description of the special processing that was done. This information is included in Appendix G.</p>															
Sol	8 (4,i)	<ul style="list-style-type: none"><li>Sol : for flight (Cruise &amp; Surface) or flight-like ground tests when the Secondary timestamp continuously increments, this field stores the 4-integer Sol (Mars solar day) included in the mosaic. This is often the last (i.e. latest) sol included as part of the product, but the specific sol used is up to the discretion of the data provider.</li><li>Year-DOY (4 alphanumeric) - This field stores two metadata items in the order:<ul style="list-style-type: none"><li>a) One alpha character in range "A-Z" to designate Earth Year portion of the UTC-like time value, representing Years 2017 to 2042</li><li>b) Three integers in range "001-365" representing Day-of-Year (DOY)</li></ul></li></ul> <table><tr><th colspan="2">Time Type</th><th>Values</th><th>Valid Time Range</th></tr><tr><td rowspan="2">Sol</td><td>Cruise</td><td>0000</td><td>NULL</td></tr><tr><td>Surface</td><td>0001 – 9999</td><td>1 thru 9999</td></tr><tr><td colspan="2"></td><td>— — — —</td><td>Value is</td></tr></table>	Time Type		Values	Valid Time Range	Sol	Cruise	0000	NULL	Surface	0001 – 9999	1 thru 9999			— — — —	Value is
Time Type		Values	Valid Time Range														
Sol	Cruise	0000	NULL														
	Surface	0001 – 9999	1 thru 9999														
		— — — —	Value is														

This document has been reviewed and determined not to contain export controlled technical data.

## Mosaic Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE		VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION	GEOID CORR	COORD FRAME	BRT CORR									PRODUCER	VERSION	.				EXT		

Field	Position (size, type)	Description										
		<div><div><div>Year-DOY</div><div>Cruise/Surface Test</div></div><div><div>A001,...A365</div><div>.....</div><div>Z001,Z002,...Z365</div></div><div><div>out of range</div><div>2017 DOY 1-365</div><div>.....</div><div>2042 DOY 1-365</div></div></div>										
Multisol	12 (1,a)	Flag indicating that a significant percentage of the data content was acquired on more than a single Sol, i.e. across multiple Sols. Specification of this flag is at the discretion of the mosaic product's producer, who determines what percentage is "significant". Valid values: <ul style="list-style-type: none"><li>_ : Flags mosaic as single sol (insignificant multiple sol contribution)</li><li>X : Flags mosaic as containing significant content from multiple sols</li></ul>										
Product Type	13 (3,a)	Product Identifier. See Table 17-2 and Table 17-3 for details.										
Venue	16 (1,a)	Mission venue identifier: <ul style="list-style-type: none"><li>_ : Flight (surface or cruise)</li><li>A : AVSTB</li><li>F : FSWTB</li><li>M : MSTB</li><li>R : "ROASTT"</li><li>S : "Scarecrow"</li><li>V : VSTB</li></ul> <ul style="list-style-type: none"><li>Other venue identifiers may be defined later.</li></ul>										
Site/Flight/Set	17 (3, i/a)	Site/Flight/Set identifier; first element of the RMC (see Section 4.4).  Site location count from the RMC where the data was acquired for the rover. Flight number from the RMC where the data was acquired for the helicopter. Set number from the RMC where the data was acquired for LCAM (always 0). <table><tr><th>Values</th><th>Range</th></tr><tr><td>000, 001, ..., 999</td><td>0 thru 999</td></tr><tr><td>A00, A01, ..., A99</td><td>1000 thru 1099</td></tr><tr><td>B00, B01, ..., B99</td><td>1100 thru 1199</td></tr><tr><td>...</td><td>...</td></tr></table>	Values	Range	000, 001, ..., 999	0 thru 999	A00, A01, ..., A99	1000 thru 1099	B00, B01, ..., B99	1100 thru 1199	...	...
Values	Range											
000, 001, ..., 999	0 thru 999											
A00, A01, ..., A99	1000 thru 1099											
B00, B01, ..., B99	1100 thru 1199											
...	...											

This document has been reviewed and determined not to contain export controlled technical data.

## Mosaic Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE		VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION	GEOM CORR	COORD FRAME	BRT CORR									PRODUCER	VERSION	.			EXT			

Field	Position (size, type)	Description																														
		<table><tr><td>Z00, Z01, ... Z99</td><td>3500 thru 3599</td></tr><tr><td>AA0, AA1, ..., AA9</td><td>3600 thru 3609</td></tr><tr><td>AB0, AB1, ..., AB9</td><td>3610 thru 3619</td></tr><tr><td>...</td><td>...</td></tr><tr><td>ZZ0, ZZ1, ..., ZZ9</td><td>10350 thru 10359</td></tr><tr><td>AAA, AAB, ..., AAZ</td><td>10360 thru 10385</td></tr><tr><td>ABA, ABB, ..., ABZ</td><td>10386 thru 10411</td></tr><tr><td>...</td><td>...</td></tr><tr><td>ZZA, ZZB, ..., ZZZ</td><td>27910 thru 27935</td></tr><tr><td>0AA, 0AB, ..., 0AZ</td><td>27936 thru 27961</td></tr><tr><td>0BA, 0BB, ..., 0BZ</td><td>27962 thru 27987</td></tr><tr><td>...</td><td>...</td></tr><tr><td>7CA, 7CB, ..., 7CZ</td><td>32720 thru 32745</td></tr><tr><td>7DA, 7DB, ..., 7DV</td><td>32746 thru 32767</td></tr><tr><td>— — —</td><td>Value out of range</td></tr></table>	Z00, Z01, ... Z99	3500 thru 3599	AA0, AA1, ..., AA9	3600 thru 3609	AB0, AB1, ..., AB9	3610 thru 3619	...	...	ZZ0, ZZ1, ..., ZZ9	10350 thru 10359	AAA, AAB, ..., AAZ	10360 thru 10385	ABA, ABB, ..., ABZ	10386 thru 10411	...	...	ZZA, ZZB, ..., ZZZ	27910 thru 27935	0AA, 0AB, ..., 0AZ	27936 thru 27961	0BA, 0BB, ..., 0BZ	27962 thru 27987	...	...	7CA, 7CB, ..., 7CZ	32720 thru 32745	7DA, 7DB, ..., 7DV	32746 thru 32767	— — —	Value out of range
Z00, Z01, ... Z99	3500 thru 3599																															
AA0, AA1, ..., AA9	3600 thru 3609																															
AB0, AB1, ..., AB9	3610 thru 3619																															
...	...																															
ZZ0, ZZ1, ..., ZZ9	10350 thru 10359																															
AAA, AAB, ..., AAZ	10360 thru 10385																															
ABA, ABB, ..., ABZ	10386 thru 10411																															
...	...																															
ZZA, ZZB, ..., ZZZ	27910 thru 27935																															
0AA, 0AB, ..., 0AZ	27936 thru 27961																															
0BA, 0BB, ..., 0BZ	27962 thru 27987																															
...	...																															
7CA, 7CB, ..., 7CZ	32720 thru 32745																															
7DA, 7DB, ..., 7DV	32746 thru 32767																															
— — —	Value out of range																															
Drive/Pos/ Instance	20 (4, i/a)	<p>Drive/Pos/Instance identifier; second element of the RMC (see Section 4.4).</p> <p>Drive count (position within a Site location) from the RMC where the data was acquired for the rover.</p> <p>Pos count from the RMC where the data was acquired for the helicopter.</p> <p>Instance number from the RMC where the data was acquired for LCAM.</p> <table><tr><th>Values</th><th>Range</th></tr><tr><td>0000, 0001, ..., 9999</td><td>0 thru 9999</td></tr><tr><td>A000, A001, ..., A999</td><td>10000 thru 10999</td></tr><tr><td>B000, B001, ..., B999</td><td>11000 thru 11999</td></tr><tr><td>...</td><td>...</td></tr><tr><td>Z000, Z001, ... Z999</td><td>35000 thru 35999</td></tr><tr><td>AA00, AA01, ..., AA99</td><td>36000 thru 36099</td></tr><tr><td>AB00, AB01, ..., AB99</td><td>36100 thru 36199</td></tr><tr><td>...</td><td>...</td></tr><tr><td>AZ00, AZ01, ..., AZ99</td><td>38500 thru 38599</td></tr></table>	Values	Range	0000, 0001, ..., 9999	0 thru 9999	A000, A001, ..., A999	10000 thru 10999	B000, B001, ..., B999	11000 thru 11999	...	...	Z000, Z001, ... Z999	35000 thru 35999	AA00, AA01, ..., AA99	36000 thru 36099	AB00, AB01, ..., AB99	36100 thru 36199	...	...	AZ00, AZ01, ..., AZ99	38500 thru 38599										
Values	Range																															
0000, 0001, ..., 9999	0 thru 9999																															
A000, A001, ..., A999	10000 thru 10999																															
B000, B001, ..., B999	11000 thru 11999																															
...	...																															
Z000, Z001, ... Z999	35000 thru 35999																															
AA00, AA01, ..., AA99	36000 thru 36099																															
AB00, AB01, ..., AB99	36100 thru 36199																															
...	...																															
AZ00, AZ01, ..., AZ99	38500 thru 38599																															

This document has been reviewed and determined not to contain export controlled technical data.

## Mosaic Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE		VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION	GEOM CORR	COORD FRAME	BRT CORR									PRODUCER	VERSION	.			EXT			

Field	Position (size, type)	Description
		BA00, BA01, ..., BA99      38600 thru 38699 BB00, BB01, ..., BB99      38700 thru 38799 ...                                  ... LJ00, LJ01, ..., LJ35      65500 thru 65535 - - - - -                      Value is out of range
Multidrive	24 (1,a)	Flag indicating that a significant percentage of the data content was acquired on more than a single Drive/Pos/Instance, i.e. across multiple drives. Valid values: <ul style="list-style-type: none"> <li>_ : Flags mosaic as single Drive/Pos/Instance position</li> <li>X : Flags mosaic as containing multiple Drive/Pos/Instance positions</li> </ul>
Projection	25 (3,a)	Projection type. Valid values: <ul style="list-style-type: none"> <li>CYL - Cylindrical</li> <li>CYP - Cylindrical-Perspective</li> <li>ORT - Orthographic</li> <li>ORR - Orthorectified</li> <li>PER - Perspective</li> <li>POL - Polar</li> <li>VRT - Vertical (special case of Orthographic)</li> </ul>
Geometric Correction	28 (1,a)	Geometric correction type indicator. Specifies the correction type that was applied to the largest percentage of data content. Valid values are: <ul style="list-style-type: none"> <li>_ : No correction (raw pointing)</li> <li>A : Auto-correction via tiepointing</li> <li>F : Auto-correction via tiepointing &amp; auto-registration with fiducials</li> <li>G : Auto-correction via tiepointing &amp; manual registration with fiducials</li> <li>T : Manual tiepointing</li> <li>R : Manual tiepointing &amp; auto-registration with fiducials</li> <li>M : Manual tiepointing &amp; manual registration with fiducials</li> <li>O : Other correction not listed above</li> </ul>
Coordinate Frame	29 (1,a)	Coordinate system (frame) type. Valid values: <ul style="list-style-type: none"> <li>S : Site frame</li> <li>L : Local Level frame</li> <li>R : Rover Frame</li> <li>U : Untilt (CYP only, is Rover frame with rotation)</li> </ul>

## Mosaic Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE		VENUE		SITE			DRIVE				MULTI DRIVE	PROJECTION		GEOM CORR	COORD FRAME	BRT CORR							PRODUCER	VERSION	.		EXT				

Field	Position (size, type)	Description																								
		<ul style="list-style-type: none"><li>O : Other</li></ul>																								
Brt Corr.	30 (1, a)	Brightness correction type indicator. Specifies the correction type that was applied to the largest percentage of data content. Valid values are: <ul style="list-style-type: none"><li>_ : No correction</li><li>B : Automatic brightness adjustment (multiplicative and/or additive factor applied to each frame)</li><li>M : Manual brightness adjustment (same factors as “B”)</li><li>V : Anti-vignetting adjustment applied to some or all frames</li><li>G : General brightness correction</li><li>A : General brightness correction that can vary across the frame (automatic)</li><li>O : Other correction not listed above</li></ul>																								
Description	31 (7,a)	General purpose mosaic identifier. Can be set to anything to help identify the mosaic, such as target name, panorama name, theme name, method of production, etc. Valid values include A-Z, 0-9, and underscore. Must always pad to 7 characters, using underscores as necessary.																								
Producer	38 (1,a)	Identifier for the institution/team that created this product: <ul style="list-style-type: none"><li>J : JPL (IDS/MIPL)</li><li>A: ASU (Mastcam-Z team)</li><li>P : Principal investigator of instrument.</li></ul> <table><tr><th>Instrument</th><th>PI</th></tr><tr><td>ECAM</td><td>JPL</td></tr><tr><td>ZCAM</td><td>ASU (Tempe, AZ)</td></tr><tr><td>SCAM RMI</td><td>IRAP (France)</td></tr><tr><td>PIXL MCC</td><td>JPL</td></tr><tr><td>SHERLOC</td><td>JPL</td></tr><tr><td>MEDA</td><td>CAB CSIC-INTA</td></tr><tr><td>Skycam</td><td>(Spain)</td></tr><tr><td>EDL</td><td>JPL</td></tr><tr><td>Cameras</td><td></td></tr><tr><td>HELI</td><td>JPL</td></tr><tr><td>RTE/NAV</td><td></td></tr></table> <ul style="list-style-type: none"><li>A – I, K – O, Q – Z : Co-I to be identified per instrument at the discretion of the instrument PI.</li><li>_ : undefined/other</li></ul>	Instrument	PI	ECAM	JPL	ZCAM	ASU (Tempe, AZ)	SCAM RMI	IRAP (France)	PIXL MCC	JPL	SHERLOC	JPL	MEDA	CAB CSIC-INTA	Skycam	(Spain)	EDL	JPL	Cameras		HELI	JPL	RTE/NAV	
Instrument	PI																									
ECAM	JPL																									
ZCAM	ASU (Tempe, AZ)																									
SCAM RMI	IRAP (France)																									
PIXL MCC	JPL																									
SHERLOC	JPL																									
MEDA	CAB CSIC-INTA																									
Skycam	(Spain)																									
EDL	JPL																									
Cameras																										
HELI	JPL																									
RTE/NAV																										

This document has been reviewed and determined not to contain export controlled technical data.

## Mosaic Filename Convention

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE		VENUE		SITE			DRIVE				MULTI DRIVE	PROJECTION	GEOM CORR	COORD FRAME	BRT CORR	DESCRIPTION		PRODUCER	VERSION	.									EXT		

Field	Position (size, type)	Description																				
		Other producer codes will be added in the future.																				
Version	39 (2,a)	<p>Product version number. Increments by one whenever a previously generated file with an otherwise identical filename exists.</p> <table><thead><tr><th>Values</th><th>Range</th></tr></thead><tbody><tr><td>01, 02 ..., 99</td><td>1 thru 99</td></tr><tr><td>A0, A1, ..., A9</td><td>100 thru 109</td></tr><tr><td>AA, AB, ..., AZ</td><td>110 thru 135</td></tr><tr><td>B0, B1, B2 ..., B9</td><td>136 thru 145</td></tr><tr><td>BA, BB, ..., BZ</td><td>146 thru 171</td></tr><tr><td>...</td><td>...</td></tr><tr><td>Z0, Z1, ..., Z9</td><td>1000 thru 1009</td></tr><tr><td>ZA, ZB, ..., ZZ</td><td>1010 thru 1035</td></tr><tr><td>--</td><td>Value is out of range</td></tr></tbody></table> <ul style="list-style-type: none"><li>Every version need not exist. E.g. version 01, 02, and 04 may exist but not 03. In general, the highest-numbered version represents the best version of that product. This field increments independently of all fields.</li></ul>	Values	Range	01, 02 ..., 99	1 thru 99	A0, A1, ..., A9	100 thru 109	AA, AB, ..., AZ	110 thru 135	B0, B1, B2 ..., B9	136 thru 145	BA, BB, ..., BZ	146 thru 171	...	...	Z0, Z1, ..., Z9	1000 thru 1009	ZA, ZB, ..., ZZ	1010 thru 1035	--	Value is out of range
Values	Range																					
01, 02 ..., 99	1 thru 99																					
A0, A1, ..., A9	100 thru 109																					
AA, AB, ..., AZ	110 thru 135																					
B0, B1, B2 ..., B9	136 thru 145																					
BA, BB, ..., BZ	146 thru 171																					
...	...																					
Z0, Z1, ..., Z9	1000 thru 1009																					
ZA, ZB, ..., ZZ	1010 thru 1035																					
--	Value is out of range																					
.	41 (1,a)	Separator for filename and extension. Always set to “.”																				
Extension	42 (3,a)	<p>File extension:</p> <ul style="list-style-type: none"><li>VIC : Used for VICAR files.</li><li>IMG : Same as .VIC with ODL label.</li><li>TIF : TIFF formatted image file. (no label)</li><li>JPG : JPEG formatted image file. (no label)</li><li>PNG : PNG formatted image file. (no label)</li><li>TXT : ASCII text file.</li><li>xml : PDS4 label file</li></ul>																				

The filename convention used for Unified Terrain Meshes is shown in Table 18-3 below.

### Table 18-3 – Camera instrument unified terrain mesh filenaming convention

1 to n	2n	3n	4n	5n	6n	7n	8n	9n	10n	11n	12n	13n	14n	15n	16n	17n	18n	19n	20n	21n	22n	23n	24n	25n	26n	27n	28n	29n	30n	31n	32n	33n	34n	35n	36n	37n	38n	39n	40n	41n	42n	43n	44n
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL				MULTI SOL		XYZ Type			GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE			SITE		MULTI SITE		DRIVE			MULTI DRIVE		DESCRIPTION					PRODUCER		VERSION				EXT	

Field	Position (size, type)	Description
Instrument	1+n (n,a)	First Instrument Identifier: <ul style="list-style-type: none"> <li>• N : Navcam</li> <li>• Z : Mastcam-Z</li> <li>• L : Supercam RMI</li> <li>• O: Orbital Mesh</li> <li>• P : PIXL Micro Context Cam (MCC)</li> <li>• I : SHERLOC Imaging (WATSON)</li> <li>• C : SHERLOC ACI</li> <li>• F : Front Hazcam (RCE-A)</li> <li>• B : Front Hazcam (RCE-B)</li> <li>• R : Rear Hazcam</li> <li>• E : EDL Camera</li> <li>• H : Helicopter Return to Earth Cam</li> <li>• V : Helicopter Navigation Cam</li> </ul>
_	2+n (1,a)	Underscore for readability. Always set to “_”.
Eye	3+n (1,a)	Identifies the eye of the data used to make this mesh. <ul style="list-style-type: none"> <li>• L : Left-eye</li> <li>• R : Right-eye</li> </ul>
Mesh Type	4+n (1,a)	Identifies the type of mesh. <ul style="list-style-type: none"> <li>• T : Tactical</li> <li>• C : Contextual</li> <li>• H : Helicopter</li> <li>• O : Other</li> </ul>
Special	5+n (1,a)	<p>Special Processing flag. The special processing character is used to indicate off-nominal or special processing of the mesh. Examples include use of different correlation parameters, special stretches to eliminate shadows, reprocessing with different camera pointing, etc.</p> <p>The meaning of any individual character in this field (other than "_" which means nominal processing) will be defined on an ad-hoc basis as needed during the mission. Within one Sol or a range of Sols, the character will be used consistently. So, this field can be used to group together all derived products resulting from one kind of special processing. An attempt will be made to maintain consistency across different Sols as well, but this may not always be possible; thus the meaning of characters may change across different individual or ranges</p>

This document has been reviewed and determined not to contain export controlled technical data.

## Mesh Filename Convention

1 to n	2n	3n	4n	5n	6n	7n	8n	9n	10n	11n	12n	13n	14n	15n	16n	17n	18n	19n	20n	21n	22n	23n	24n	25n	26n	27n	28n	29n	30n	31n	32n	33n	34n	35n	36n	37n	38n	39n	40n	41n	42n	43n	44n
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL	MULTI SOL	XYZ Type	GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE	SITE	MULTI SITE	DRIVE	MULTI DRIVE	DESCRIPTION	PRODUCER	VERSION	.	EXT																					

Field	Position (size, type)	Description																				
		<p>of Sols.</p> <p>A database will be maintained containing all special processing designators that are used, the sols they relate to, and a description of the special processing that was done. This information is included in Appendix G.</p>																				
Sol	6+n (4,i)	<div><ul style="list-style-type: none"><li>Sol : For flight (Cruise &amp; Surface) or flight-like ground tests when the Secondary timestamp continuously increments, this field stores a 4-digit integer that indicates one of the Sols (Mars solar day) included in the mesh. This is often the last (i.e. latest) sol included as part of the product, but the specific sol used is up to the discretion of the data provider.</li><li>Year-DOY (4 alphanumeric) - This field stores two metadata items in the order:<div><div>a) One alpha character in range “A-Z” to designate Earth Year portion of the UTC-like time value, representing Years 2017 to 2042</div><div>b) Three integers in range “001-365” representing Day-of-Year (DOY)</div></div></li></ul></div> <table><tr><th colspan="2">Time Type</th><th>Values</th><th>Valid Time Range</th></tr><tr><td rowspan="3">Sol</td><td>Cruise</td><td>0000</td><td>NULL</td></tr><tr><td>Surface</td><td>0001 – 9999</td><td>1 thru 9999</td></tr><tr><td></td><td>----</td><td>Value is out of range</td></tr><tr><td rowspan="2">Year-DOY</td><td rowspan="2">Cruise/Surface Test</td><td>A001,...A365</td><td>2017 DOY 1-365</td></tr><tr><td>..... Z001,Z002,...Z365</td><td>..... 2042 DOY 1-365</td></tr></table>	Time Type		Values	Valid Time Range	Sol	Cruise	0000	NULL	Surface	0001 – 9999	1 thru 9999		----	Value is out of range	Year-DOY	Cruise/Surface Test	A001,...A365	2017 DOY 1-365	..... Z001,Z002,...Z365	..... 2042 DOY 1-365
Time Type		Values	Valid Time Range																			
Sol	Cruise	0000	NULL																			
	Surface	0001 – 9999	1 thru 9999																			
		----	Value is out of range																			
Year-DOY	Cruise/Surface Test	A001,...A365	2017 DOY 1-365																			
		..... Z001,Z002,...Z365	..... 2042 DOY 1-365																			
Multi sol	10+n (1,a)	<p>Flag indicating that a significant percentage of the data content was acquired on more than a single Sol, i.e. across multiple Sols. Specification of this flag is at the discretion of the mosaic product's producer, who determines what percentage is "significant". Valid values:</p> <ul style="list-style-type: none"><li>_ : Flags mosaic as single sol (insignificant multiple sol contribution)</li><li>X : Flags mosaic as containing significant content from multiple sols</li></ul>																				
XYZ Type	11+n (3,a)	Indicates the XYZ type used for mesh data:																				



## Mesh Filename Convention

1 to n	2n	3n	4n	5n	6n	7n	8n	9n	10n	11n	12n	13n	14n	15n	16n	17n	18n	19n	20n	21n	22n	23n	24n	25n	26n	27n	28n	29n	30n	31n	32n	33n	34n	35n	36n	37n	38n	39n	40n	41n	42n	43n	44n		
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL	MULTI SOL	XYZ Type	GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE	SITE	MULTI SITE	DRIVE	MULTI DRIVE	DESCRIPTION	PRODUCER	VERSION	.	EXT																							

Field	Position (size, type)	Description
		<ul style="list-style-type: none"> <li>See Table 6.3.1.2</li> </ul>
Geometry	14+n (1,a)	<p>Linearization flag:</p> <ul style="list-style-type: none"> <li>_ : Non-linearized (raw geometry)</li> <li>L : Product has been linearized with nominal stereo partner</li> <li>A : Product has been linearized with an actual stereo partner.</li> <li>T: Product has had trapezoidal correction applied (PIXL MCC)</li> </ul> <p>Note that for the “A” case, an image can have multiple stereo partners and the linearized images will be different for each partner. A user will need to look in the ODL/VICAR label to determine which partner was used for linearization.</p>
Frame	15+n (1,a)	<p>Coordinate system (frame) type. Valid values:</p> <ul style="list-style-type: none"> <li>S : Site frame</li> <li>L : Local Level frame</li> <li>R : Rover Frame</li> <li>O : Other</li> </ul>
Resolution	16+n (1,i/a)	<p>Indicates if the input data is comprised of single or multi-resolution images (ECAM reconstructed). Number value indicates level of pixel averaging.</p> <ul style="list-style-type: none"> <li>1 : 1x1</li> <li>2 : 2x2</li> <li>4 : 4x4</li> <li>8 : 8x8</li> <li>M : multi-resolution</li> </ul>
Pyramid	17+n (1,i)	<p>Indicates the mesh pyramid level or downsampling relative to the highest resolution of the mesh. (powers of 2).</p> <ul style="list-style-type: none"> <li>0 : full resolution</li> <li>1 : 2x downsampling</li> <li>2 : 4x downsampling</li> <li>3 : 8x downsampling</li> <li>4 : 16x downsampling</li> <li>n : 2<sup>n</sup> downsampling</li> </ul>
Venue	18+n (1,a)	<p>Mission venue identifier:</p> <ul style="list-style-type: none"> <li>_ : Flight (surface or cruise)</li> <li>A : AVSTB</li> <li>F : FSWTB</li> <li>M : MSTB</li> <li>R : “ROASTT”</li> <li>S : “Scarecrow”</li> </ul>

## Mesh Filename Convention

1 to n	2 <sub>n</sub>	3 <sub>n</sub>	4 <sub>n</sub>	5 <sub>n</sub>	6 <sub>n</sub>	7 <sub>n</sub>	8 <sub>n</sub>	9 <sub>n</sub>	10 <sub>n</sub>	11 <sub>n</sub>	12 <sub>n</sub>	13 <sub>n</sub>	14 <sub>n</sub>	15 <sub>n</sub>	16 <sub>n</sub>	17 <sub>n</sub>	18 <sub>n</sub>	19 <sub>n</sub>	20 <sub>n</sub>	21 <sub>n</sub>	22 <sub>n</sub>	23 <sub>n</sub>	24 <sub>n</sub>	25 <sub>n</sub>	26 <sub>n</sub>	27 <sub>n</sub>	28 <sub>n</sub>	29 <sub>n</sub>	30 <sub>n</sub>	31 <sub>n</sub>	32 <sub>n</sub>	33 <sub>n</sub>	34 <sub>n</sub>	35 <sub>n</sub>	36 <sub>n</sub>	37 <sub>n</sub>	38 <sub>n</sub>	39 <sub>n</sub>	40 <sub>n</sub>	41 <sub>n</sub>	42 <sub>n</sub>	43 <sub>n</sub>	44 <sub>n</sub>			
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL	MULTI SOL	XYZ Type	GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE	SITE	MULTI SITE	DRIVE	MULTI DRIVE	DESCRIPTION	PRODUCER	VERSION	.	EXT																								

Field	Position (size, type)	Description																																								
		<ul style="list-style-type: none"><li>V : VSTB</li></ul> <p>Other venue identifiers may be defined later.</p>																																								
Texture Type	19+n (3,a)	Indicates the product type used for the mesh skin: <ul style="list-style-type: none"><li>See Table 6.3.1.2</li><li>___: no mesh skin</li></ul>																																								
Site/Flight/Set	22+n (3,i/a)	<p>Site/Flight/Set identifier; first element of the RMC (see Section 4.4).</p> <p>Site location count from the RMC where the data was acquired for the rover. Flight number from the RMC where the data was acquired for the helicopter. Set number from the RMC where the data was acquired for LCAM (always 0).</p> <table><thead><tr><th>Values</th><th>Range</th></tr></thead><tbody><tr><td>000, 001, ..., 999</td><td>0 thru 999</td></tr><tr><td>A00, A01, ..., A99</td><td>1000 thru 1099</td></tr><tr><td>B00, B01, ..., B99</td><td>1100 thru 1199</td></tr><tr><td>...</td><td>...</td></tr><tr><td>Z00, Z01, ... Z99</td><td>3500 thru 3599</td></tr><tr><td>AA0, AA1, ..., AA9</td><td>3600 thru 3609</td></tr><tr><td>AB0, AB1, ..., AB9</td><td>3610 thru 3619</td></tr><tr><td>...</td><td>...</td></tr><tr><td>ZZ0, ZZ1, ..., ZZ9</td><td>10350 thru 10359</td></tr><tr><td>AAA, AAB, ..., AAZ</td><td>10360 thru 10385</td></tr><tr><td>ABA, ABB, ..., ABZ</td><td>10386 thru 10411</td></tr><tr><td>...</td><td>...</td></tr><tr><td>ZZA, ZZB, ..., ZZZ</td><td>27910 thru 27935</td></tr><tr><td>0AA, 0AB, ..., 0AZ</td><td>27936 thru 27961</td></tr><tr><td>0BA, 0BB, ..., 0BZ</td><td>27962 thru 27987</td></tr><tr><td>...</td><td>...</td></tr><tr><td>7CA, 7CB, ..., 7CZ</td><td>32720 thru 32745</td></tr><tr><td>7DA, 7DB, ..., 7DV</td><td>32746 thru 32767</td></tr><tr><td>---</td><td>Value out of range</td></tr></tbody></table>	Values	Range	000, 001, ..., 999	0 thru 999	A00, A01, ..., A99	1000 thru 1099	B00, B01, ..., B99	1100 thru 1199	...	...	Z00, Z01, ... Z99	3500 thru 3599	AA0, AA1, ..., AA9	3600 thru 3609	AB0, AB1, ..., AB9	3610 thru 3619	...	...	ZZ0, ZZ1, ..., ZZ9	10350 thru 10359	AAA, AAB, ..., AAZ	10360 thru 10385	ABA, ABB, ..., ABZ	10386 thru 10411	...	...	ZZA, ZZB, ..., ZZZ	27910 thru 27935	0AA, 0AB, ..., 0AZ	27936 thru 27961	0BA, 0BB, ..., 0BZ	27962 thru 27987	...	...	7CA, 7CB, ..., 7CZ	32720 thru 32745	7DA, 7DB, ..., 7DV	32746 thru 32767	---	Value out of range
Values	Range																																									
000, 001, ..., 999	0 thru 999																																									
A00, A01, ..., A99	1000 thru 1099																																									
B00, B01, ..., B99	1100 thru 1199																																									
...	...																																									
Z00, Z01, ... Z99	3500 thru 3599																																									
AA0, AA1, ..., AA9	3600 thru 3609																																									
AB0, AB1, ..., AB9	3610 thru 3619																																									
...	...																																									
ZZ0, ZZ1, ..., ZZ9	10350 thru 10359																																									
AAA, AAB, ..., AAZ	10360 thru 10385																																									
ABA, ABB, ..., ABZ	10386 thru 10411																																									
...	...																																									
ZZA, ZZB, ..., ZZZ	27910 thru 27935																																									
0AA, 0AB, ..., 0AZ	27936 thru 27961																																									
0BA, 0BB, ..., 0BZ	27962 thru 27987																																									
...	...																																									
7CA, 7CB, ..., 7CZ	32720 thru 32745																																									
7DA, 7DB, ..., 7DV	32746 thru 32767																																									
---	Value out of range																																									

## Mesh Filename Convention

1 to n	2 <sub>n</sub>	3 <sub>n</sub>	4 <sub>n</sub>	5 <sub>n</sub>	6 <sub>n</sub>	7 <sub>n</sub>	8 <sub>n</sub>	9 <sub>n</sub>	10 <sub>n</sub>	11 <sub>n</sub>	12 <sub>n</sub>	13 <sub>n</sub>	14 <sub>n</sub>	15 <sub>n</sub>	16 <sub>n</sub>	17 <sub>n</sub>	18 <sub>n</sub>	19 <sub>n</sub>	20 <sub>n</sub>	21 <sub>n</sub>	22 <sub>n</sub>	23 <sub>n</sub>	24 <sub>n</sub>	25 <sub>n</sub>	26 <sub>n</sub>	27 <sub>n</sub>	28 <sub>n</sub>	29 <sub>n</sub>	30 <sub>n</sub>	31 <sub>n</sub>	32 <sub>n</sub>	33 <sub>n</sub>	34 <sub>n</sub>	35 <sub>n</sub>	36 <sub>n</sub>	37 <sub>n</sub>	38 <sub>n</sub>	39 <sub>n</sub>	40 <sub>n</sub>	41 <sub>n</sub>	42 <sub>n</sub>	43 <sub>n</sub>	44 <sub>n</sub>			
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL	MULTI SOL	XYZ Type	GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE	SITE	MULTI SITE	DRIVE	MULTI DRIVE	DESCRIPTION	PRODUCER	VERSION	.	EXT																								

Field	Position (size, type)	Description																														
Multisite	25+n (1,a)	Flag indicating that a significant percentage of the data content was acquired on more than a single Site/Flight position, i.e. across multiple sites. Valid values: <ul style="list-style-type: none"><li>_ : Flags mesh as single Site/Flight</li><li>X : Flags mesh as containing content from multiple Sites/Flights</li></ul>																														
Drive	26+n (4, i/a)	<div>Drive/Pos/Instance identifier; second element of the RMC (see Section 4.4).</div> <div>Drive count (position within a Site location) from the RMC where the data was acquired for the rover.</div> <div>Pos count from the RMC where the data was acquired for the helicopter.</div> <div>Instance number from the RMC where the data was acquired for LCAM.</div> <table><thead><tr><th>Values</th><th>Range</th></tr></thead><tbody><tr><td>0000, 0001, ..., 9999</td><td>0 thru 9999</td></tr><tr><td>A000, A001, ..., A999</td><td>10000 thru 10999</td></tr><tr><td>B000, B001, ..., B999</td><td>11000 thru 11999</td></tr><tr><td>...</td><td>...</td></tr><tr><td>Z000, Z001, ... Z999</td><td>35000 thru 35999</td></tr><tr><td>AA00, AA01, ..., AA99</td><td>36000 thru 36099</td></tr><tr><td>AB00, AB01, ..., AB99</td><td>36100 thru 36199</td></tr><tr><td>...</td><td>...</td></tr><tr><td>AZ00, AZ01, ..., AZ99</td><td>38500 thru 38599</td></tr><tr><td>BA00, BA01, ..., BA99</td><td>38600 thru 38699</td></tr><tr><td>BB00, BB01, ..., BB99</td><td>38700 thru 38799</td></tr><tr><td>...</td><td>...</td></tr><tr><td>LJ00, LJ01, ..., LJ35</td><td>65500 thru 65535</td></tr><tr><td>----</td><td>Value is out of range</td></tr></tbody></table>	Values	Range	0000, 0001, ..., 9999	0 thru 9999	A000, A001, ..., A999	10000 thru 10999	B000, B001, ..., B999	11000 thru 11999	...	...	Z000, Z001, ... Z999	35000 thru 35999	AA00, AA01, ..., AA99	36000 thru 36099	AB00, AB01, ..., AB99	36100 thru 36199	...	...	AZ00, AZ01, ..., AZ99	38500 thru 38599	BA00, BA01, ..., BA99	38600 thru 38699	BB00, BB01, ..., BB99	38700 thru 38799	...	...	LJ00, LJ01, ..., LJ35	65500 thru 65535	----	Value is out of range
Values	Range																															
0000, 0001, ..., 9999	0 thru 9999																															
A000, A001, ..., A999	10000 thru 10999																															
B000, B001, ..., B999	11000 thru 11999																															
...	...																															
Z000, Z001, ... Z999	35000 thru 35999																															
AA00, AA01, ..., AA99	36000 thru 36099																															
AB00, AB01, ..., AB99	36100 thru 36199																															
...	...																															
AZ00, AZ01, ..., AZ99	38500 thru 38599																															
BA00, BA01, ..., BA99	38600 thru 38699																															
BB00, BB01, ..., BB99	38700 thru 38799																															
...	...																															
LJ00, LJ01, ..., LJ35	65500 thru 65535																															
----	Value is out of range																															
Multidrive	30+n (1,a)	Flag indicating that a significant percentage of the data content was acquired on more than a single Drive/Pos/Instance, i.e. across multiple drives. Valid values: <ul style="list-style-type: none"><li>_ : Flags mesh as single Drive/Pos/Instance position</li><li>X : Flags mesh as containing multiple Drive/Pos/Instance positions</li></ul>																														
Description	31+n (7,a)	General purpose mesh identifier. Can be set to anything to help identify the mosaic, such as target name, panorama name, theme name, method of production, etc. Valid values include A-Z, 0-9, and underscore. Must always pad to 7 characters, using underscores as necessary.																														

## Mesh Filename Convention

1 to n	2 <sub>n</sub>	3 <sub>n</sub>	4 <sub>n</sub>	5 <sub>n</sub>	6 <sub>n</sub>	7 <sub>n</sub>	8 <sub>n</sub>	9 <sub>n</sub>	10 <sub>n</sub>	11 <sub>n</sub>	12 <sub>n</sub>	13 <sub>n</sub>	14 <sub>n</sub>	15 <sub>n</sub>	16 <sub>n</sub>	17 <sub>n</sub>	18 <sub>n</sub>	19 <sub>n</sub>	20 <sub>n</sub>	21 <sub>n</sub>	22 <sub>n</sub>	23 <sub>n</sub>	24 <sub>n</sub>	25 <sub>n</sub>	26 <sub>n</sub>	27 <sub>n</sub>	28 <sub>n</sub>	29 <sub>n</sub>	30 <sub>n</sub>	31 <sub>n</sub>	32 <sub>n</sub>	33 <sub>n</sub>	34 <sub>n</sub>	35 <sub>n</sub>	36 <sub>n</sub>	37 <sub>n</sub>	38 <sub>n</sub>	39 <sub>n</sub>	40 <sub>n</sub>	41 <sub>n</sub>	42 <sub>n</sub>	43 <sub>n</sub>	44 <sub>n</sub>		
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL	MULTI SOL	XYZ Type	GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE	SITE	MULTI SITE	DRIVE	MULTI DRIVE	DESCRIPTION	PRODUCER	VERSION	.	EXT																							

Field	Position (size, type)	Description																				
Producer	38+n (1,a)	Identifier for the institution/team that created this product: <ul style="list-style-type: none"><li>J : JPL (IDS/MIPL)</li><li>A: ASU (Mastcam-Z team)</li></ul> Other producer codes will be added in the future.																				
Version	39+n (2,a)	Product version number. Increments by one whenever a previously generated file with an otherwise identical filename exists. <table><thead><tr><th>Values</th><th>Range</th></tr></thead><tbody><tr><td>01, 02 ..., 99</td><td>1 thru 99</td></tr><tr><td>A0, A1, ..., A9</td><td>100 thru 109</td></tr><tr><td>AA, AB, ..., AZ</td><td>110 thru 135</td></tr><tr><td>B0, B1, B2 ..., B9</td><td>136 thru 145</td></tr><tr><td>BA, BB, ..., BZ</td><td>146 thru 171</td></tr><tr><td>...</td><td>...</td></tr><tr><td>Z0, Z1, ..., Z9</td><td>1000 thru 1009</td></tr><tr><td>ZA, ZB, ..., ZZ</td><td>1010 thru 1035</td></tr><tr><td>--</td><td>Value is out of range</td></tr></tbody></table> <ul style="list-style-type: none"><li>Every version need not exist. E.g. version 01, 02, and 04 may exist but not 03. In general, the highest-numbered version represents the best version of that product. This field increments independently of all fields.</li></ul>	Values	Range	01, 02 ..., 99	1 thru 99	A0, A1, ..., A9	100 thru 109	AA, AB, ..., AZ	110 thru 135	B0, B1, B2 ..., B9	136 thru 145	BA, BB, ..., BZ	146 thru 171	...	...	Z0, Z1, ..., Z9	1000 thru 1009	ZA, ZB, ..., ZZ	1010 thru 1035	--	Value is out of range
Values	Range																					
01, 02 ..., 99	1 thru 99																					
A0, A1, ..., A9	100 thru 109																					
AA, AB, ..., AZ	110 thru 135																					
B0, B1, B2 ..., B9	136 thru 145																					
BA, BB, ..., BZ	146 thru 171																					
...	...																					
Z0, Z1, ..., Z9	1000 thru 1009																					
ZA, ZB, ..., ZZ	1010 thru 1035																					
--	Value is out of range																					
.	40+n (1,a)	Separator for filename and extension. Always set to “.”																				
Extension	41+n (3,a)	File extension: <ul style="list-style-type: none"><li>iv</li><li>ht</li><li>mod</li><li>mtl</li><li>obj</li></ul>																				

## 19. FILENAME MATCHING AND VERSIONING

This section discusses some nuances regarding versioning for the single-frame filename convention (Section 18.1). It does not apply to the mosaic or mesh names.

The 2020 filename is much more complex than previous missions such as MSL or MER. It is not as easy to tell which files are the “same” image as before. This is especially true for the EECAM-based cameras (Nav, Haz, CacheCam) due to tiling, but other cameras have some of the same issues.

### 19.1 Version-Like Fields

There are several fields that have characteristics similar to a version field:

- **Version.** This starts at 01 and counts up for every otherwise-identical file. Note though that if one of the other version-like fields increments, version resets: it really is used only when there are no other filename differences.
- **Reconstruction counter** (R in camera-specific field, ECAM only). This indicates when a tile reconstruction (Section 6.2.1) was updated in a manner that changed the geometry or color state of the image.
- **Stereo partner counter** (S in camera-specific field, all but PIXL). This differentiates images that have been processed in stereo with stereo partners other than the nominal one (nominal meaning simultaneous L/R exposure with the same SCLK). It does not say what the partner is (that’s in the label), and the partner need not have the same S value. A difference in this field simply indicates a difference in stereo partner. Unlike the other version-like fields, there is no sense of ordering or “better” in different S values. See Section 8.1.
- **Compression.** This field changes if the image is downlinked with a different compression. It is “version-like” in that the same image may be downlinked with different compressions, and the “highest” (least compressed) is the one that should generally be used.

### 19.2 Determining Instances of the “Same” Image

There are instances where you want to identify images as different versions of the same base. For example, only one of the “same” images needs to go through the coregistration process or be matched for stereo, or be displayed in Marsviewer. This is not saying the images are identical; they could have different resolution or compression or be retransmission of partial images. “Same” in this context means they are simply different versions of the same exposure and only one of the set generally need be processed. The one that should be processed is the “best”, which is generally determined by the sort order described in Section 19.3.

When determining “same”, the following filename fields should be ignored:

- **Color/filter.** The Bayer state may change (E=encoded -> F=full color) if de-Bayering is applied. It may also change as the result of RDR processing. By the time you get to FDR you should not see E, so there really shouldn’t be multiple values of this at the FDR stage. However, the filter may change for RDRs.
- **Special.** It indicates different processing, but of the “same” image. Generally stick with the same special ID as your inputs, or “\_” if not available.
- **Product type.** The type code varies based on processing.

This document has been reviewed and determined not to contain export controlled technical data.

- Geometry. A linearized product is the “same” as a raw product in this context. Raw is generally preferred over linearized when possible.
- Stereo partner counter (S in camera-specific). The “same” image can be matched with multiple stereo partners. The RDRs may be very different due to the different partner, but it’s still the “same” image.
- Single vs. multi-resolution reconstruction (D in camera-specific). Part of version, next section.
- Reconstruction counter (R in camera-specific). Part of version, next section.
- Downsample. This normally won’t change except for multi-resolution reconstructions.
- Compression. Mastcam-Z will likely be retransmitted with different compressions frequently. Watson and ACI may also be retransmitted with different compressions.
- Producer. The image should be the “same” regardless of producer.
- Version. Remember that version resets to 01 any time *anything* else in the filename changes. So if any of the fields above change, the version is reset. That means you really need to follow the algorithm described in the next section rather than just looking at the version.

The remaining fields should be identical for the “same” image:

- Instrument
- All 3 timestamps
- Venue
- Thumbnail
- Site
- Drive
- Sequence/RTT
- Tile number (ECAM: TT in camera-specific)
- Observation counter (RMI: NNN in camera-specific)
- PMC (MCC: PPPP in camera-specific)
- Focal length (ZCAM: ZZZ in camera-specific)

Note that many of these fields can’t change without the SCLK changing. Specifically, venue, site, drive, sequence, observation counter, and focal length cannot change for a given SCLK. However, it should be noted that the PMC (PPPP) is required for PIXL MCC image uniqueness, which means special-casing PIXL when determining “same”-ness. The PPPP has to match to be the “same”, unlike with other cameras. Also the **ECAM** tile number (TT) has to match when determining “same”-ness, if looking at ECAM tiles.

## 19.3 Sort Order for Version-like Fields

Given a set of files determined to be the “same” per the above section, how do we determine which is “best”? The last 8 characters of the filename are designed such that a simple ASCII ascending sort will provide the “best” image (“best” being the highest/last in the sort). These last 8 characters are:

- ECAM Reconstructed: DRWCCPVV
- ECAM Tile: T\_WCCPVV
- Other: xxWCCPVV

Where D=reconstruction type, R=reconstruction counter, W=downsample, CC=compression, P=producer, VV=version, T=tile counter (partial), xx=other.

In all but three of those subfields, the highest value according to standard ascending ASCII sort order is defined as the “best” individually within that field. The fields are ordered according to priority in order to support the “last 8” sorting.

Subfield W (Downsample) sorts “backwards”, with highest resolution first and lowest resolution last. Fortunately that doesn’t interfere with the sorting, as described below.

Subfield CC (Compression) sorts properly within both the JPEG and ICER compression classes, in that a higher sort order means better fidelity (less compression). Within the Lossless class, it does not matter, as there is no difference in the data due to lossless compression. Comparing classes, JPEG < ICER < Lossless, which is generally the desired order. It is possible that a low-fidelity ICER takes precedence over a JPEG 100, but that is an unlikely corner case, as these compression types are rarely if ever mixed.

Subfield P (Producer) is assumed to be constant in terms of sort order, as it will be for pipeline-generated products. There is no ordering implied amongst different producers.

Note that we assume thumbnails have been excluded from the sort - they have a higher W (downsample) subfield, which would sort as best in some cases if not excluded.

This is how sorting works in the various cases:

- ECAM Multi-resolution reconstruction: Here D=“M”, so R becomes the primary sort key. W will be constant for a given R by definition, thus it is not relevant for sorting. CC might change, but it is based on the best compression of any component tile. So if it does change, and R does not, it means better compression was added, so sort is still proper. If all else is equal, then version distinguishes. If there is no multi-resolution reconstruction, we fall back to the single-resolution case below.
- ECAM Single-resolution reconstruction: Here D=“A”, so multi-resolution reconstructions “M” are preferred if they exist. Assuming the same R, then the next search key is W, which sorts backwards. However, if there is no multi-resolution reconstruction, then by definition there is only one single-resolution reconstruction, so there should only be one value of W, thus the backward sort order of W does not matter. Otherwise, R, CC, and VV work as with multi-resolution. If you want a specific resolution, you must search for D=“A” and W=the desired value, then everything else should sort normally.
- ECAM Tile: These all contain a digit for the first character (T) so are “worse” than any reconstruction in terms of sort order. Looking at the best for an individual tile, T is constant for a given tile, so it plays no part in the sort. W is also constant for a given tile. This leaves us sorting on CC and VV, as above.
- Non-ECAM: All the other cameras use the xx field in a way that is constant per image - it doesn't vary in a version-like way. So we're left with W (also constant), thus CC and VV as above.

Note that thumbs need to be excluded from the sort: they have a higher W field so would actually take precedence in non-reconstruction cases.

## 19.4 Version Consistency between Ops and PDS

A very important property of this data set is that the version numbers used for the PDS archive are completely consistent with those used in operations. This facilitates scientific traceability. Scientific analysis by the Mars 2020 science team starts as soon as data is downlinked. Results are not “held” pending delivery of the data to PDS – publishing schedules are independent of PDS deliveries. Therefore, papers are published using references to ops data, as it is often the only data available at the time.

If version numbers were reset when the product went to archive, then there would be no reliable way to distinguish, say, ops version 1 from archive version 1. They could in fact be very different products, which could lead to misinterpretation of scientific results or inability to replicate them.

Ensuring version numbers are consistent across ops and PDS solves this problem. If someone publishes a paper using a filename from ops, that file cannot be confused with a different product that exists in the archive. The ops file may or may not be available in the archive, but it will be clear to users if the file is NOT in the archive – they will not be able to confuse it with a different file in the archive using the same version.

It is important to note that not all ops versions go to archive. Generally only the latest version available at the time of archive is delivered. So there is no guarantee that the ops product used in the publication is available to the community – ensuring it is is the responsibility of the paper author. However, the traceability is at least clear – there can be no confusion regarding any given product. If it has not been archived, the user can definitely *know* that is the case, even if the product is not available.

This naturally leads to skipped version numbers in the archive. The archive might have versions 2 and 5 only, with no version 1, 3, or 4. This is somewhat at odds with historical PDS usage where a separate versioning namespace was used for PDS than from ops, but it is the only way to ensure that users can properly trace scientific results to their source products.

Note that even the full set of ops products may skip version numbers, due to the way the pipeline is designed. Version numbers are assigned before processing, but if a new version of a product comes in before the old version has completed processing, the older versions will be suppressed by the pipeline, since they will be superseded by the newer versions. While this is not frequent, it does occur. So even in operations, users should not assume that all versions of any product exist. In particular, it is possible that there is no version 01 of a given product.



## 20. COORDINATE FRAME STANDARDS

The M2020 Frame Manager defines several dozen coordinate frames, which can be used for commanding pointing among other things. Refer to the Pointing, Positioning, Phasing and Coordinate Systems (3PCS) document [FDD\_RAPP] or the Surface Attitude, Positioning and Pointing (SAPP) Functional Design Description [FDD\_SAPP] for more details on all these coordinate frames. Only a few of them are used by the products and processes described by this SIS. This subset is described in detail in this section. The only place in this SIS where the full set of frames can appear is in the INSTRUMENT\_COORD\_FRAME\_ID label, which is a command echo.

A subset of these frames needed for a specific image or data set are defined by the

\* \_COORDINATE\_SYSTEM groups.

Note that the PLACES database (see Section 15.3) maintains both telemetered and re-localized versions of the Site and Rover Nav frames at every available index.

**Table 20-1 – Coordinate frames used for M2020 surface operations**

Frame Name (Label Keyword Value)	Short Name (SAPP FDD)	Reference Frame (Used to Define)	Coordinate Frame	
			Origin	Orientation
ROVER_NAV_FRAME	RNAV	Enclosing SITE_FRAME	Attached to rover	Aligned with rover
ROVER_MECH_FRAME	RMECH	ROVER_NAV_FRAME	Attached to rover	Aligned with rover
LOCAL_LEVEL_FRAME	LL	Enclosing SITE_FRAME	Attached to rover (coincident with Rover Nav Frame)	North/East/Nadir
SITE_FRAME	SITE(n)	Previous SITE_FRAME	Attached to surface	North/East/Nadir
RSM_HEAD_FRAME	RSM_HEAD	ROVER_NAV_FRAME	Attached to mast head	Aligned with pointing of mast head. This corresponds to RSM_HEAD in the Frame Manager
Arm Frames: ARM_BASE_FRAME ARM_TURRET_FRAME ARM_DRILL_FRAME ARM_GDRT_FRAME ARM_TOOL_FRAME ARM_WATSON_FRAME ARM_SHERLOC_FRAME ARM_PIXL_FRAME PIXL_BASE_FRAME PIXL_SENSOR_FRAME ARM_DOCKING_POST_FRAME ARM_FCS_FRAME ARM_CUSTOM_TCP_FRAME	Arm Frames: RA_BASE TURRET DRILL GDRT TOOL WATSON SHERLOC PIXL PIXL_BASE PIXL_SENSOR DOCKING_POST FCS CUSTOM_TCP	ROVER_NAV_FRAME  (PIXL_SENSOR ref = PIXL_BASE; PIXL_BASE ref = ARM_PIXL)	Attached to the tool; see 3PCS for the specific tool frame.	Aligned with tool in some way; see 3PCS [FDD_RAPP] for the specific tool Frame.
Helicopter Frames: HELI_G_FRAME HELI_M_FRAME HELI_S1_FRAME HELI_S2_FRAME				
Lander Vision System Camera: CINT_FRAME MCMF_FRAME				

## 20.1 Coordinate Frame Indices

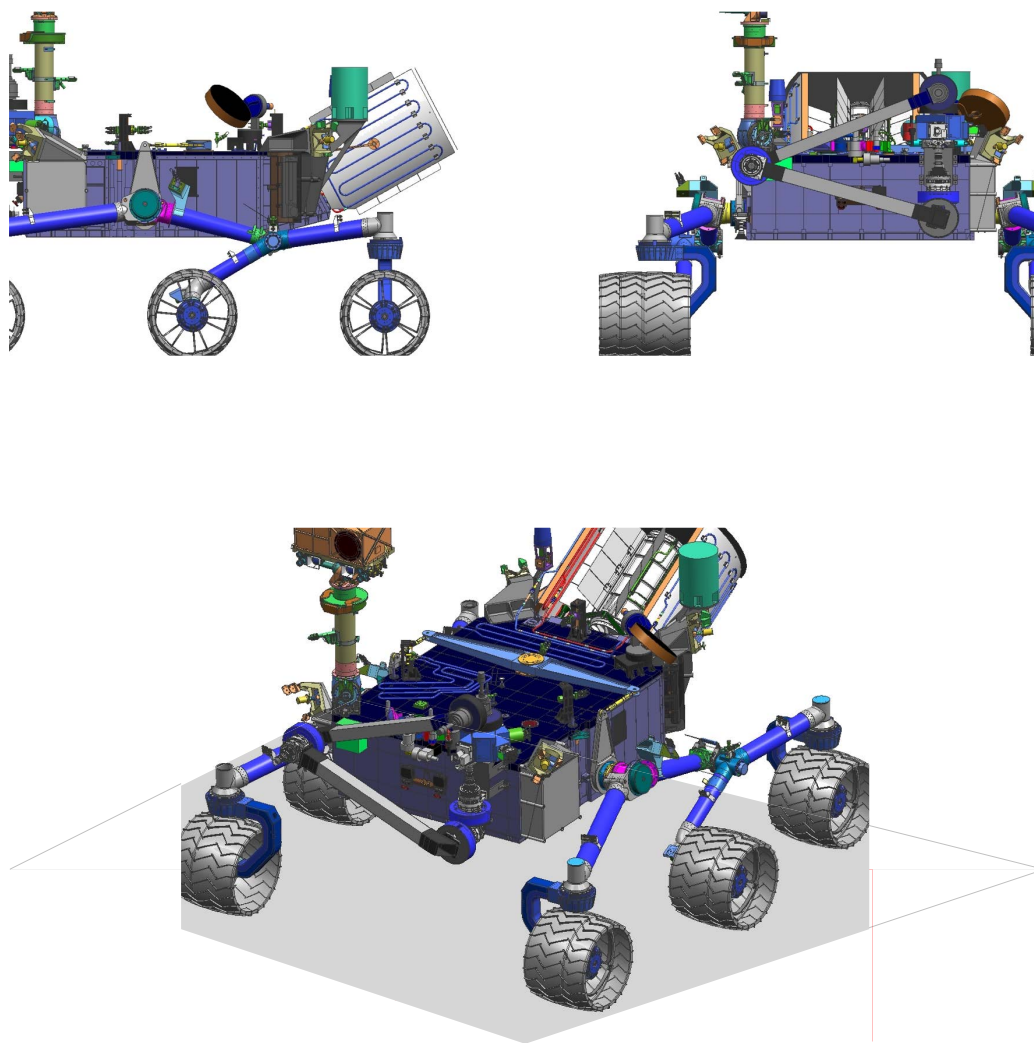
Most of the coordinate frames used by Mars 2020 are indexed; that is, there are multiple instances of the frame, identified by a set of numbers. For example, the Rover Nav frame is attached to the rover and moves with the rover, so something attached to the rover (say, a fiducial mark) is always at the same Rover Nav coordinate. No instance need be identified. But, the rover moves, so if one wanted to relate the position of that fiducial mark to a Site frame, or to latitude/longitude, the position and orientation of the Rover Nav frame at that particular point of time would have to be known. This is what is meant by an “instance” of a coordinate frame. The Rover Nav frame is defined in terms of the rover itself, but it has a position and orientation with respect to a “parent” frame, in a tree hierarchy. So if one says the fiducial is at  $(x=1, y=0.3, z=-0.8)$  in Rover Nav frame, that defines where it is on the rover at all times, but not where the fiducial was on the planet at any given time. To do that, a specific time must be provided, in order to determine the rover’s pose at that moment. This time is specified using the Rover Motion Counter (RMC, Section 4.4) and becomes an “index” to this instance of the coordinate frame. So if one additionally specifies e.g. the fiducial is at  $(x=1, y=0.3, z=-0.8)$  in Rover Nav(5,20) frame, then that coordinate could be converted to Site frame or latitude/longitude or any other coordinate frame.

Coordinate frames are distinct from Rover Motion Counter values, but the RMC value is used to identify the instance in time where the coordinate definition holds.

It should be noted that Rover Nav(5,20) frame exists regardless of whether the rover is still there or not. It is defined at the time the rover is at RMC(5,20) to be coincident with the rover and aligned with the rover, but it retains that same position and orientation in space when the rover moves on.

## 20.2 Rover Navigation (Rover Nav) Frame

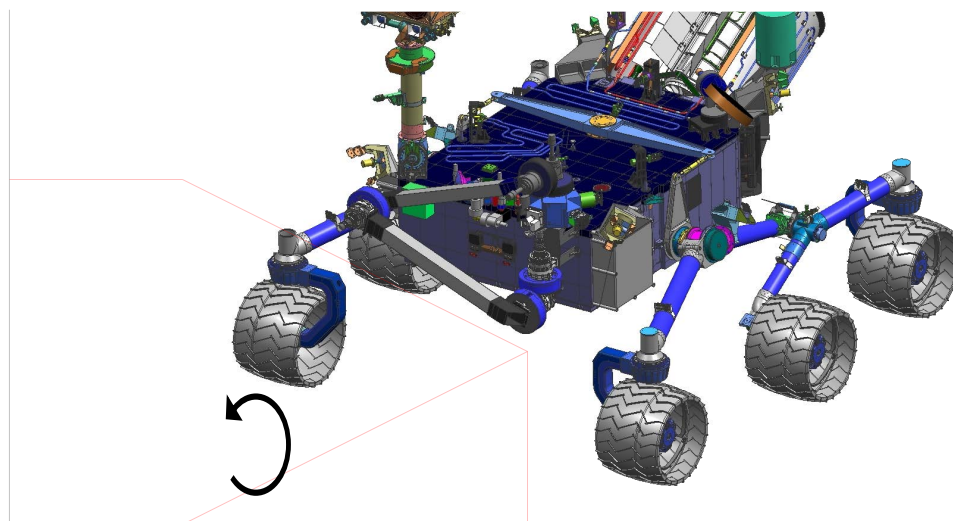
The Rover Nav frame (RNAV) is the one used for surface navigation and mobility. By definition, the frame is attached to the rover, and moves with it when the rover moves while on the surface. Its Y origin is centered on the rover and the X origin is aligned with the middle wheels’ rotation axis for the deployed rover and suspension system on a flat plane. The Z origin is defined to be at the nominal surface, which is a fixed position with respect to the rover body. The actual surface will likely not be at exactly  $Z=0$  due to the effects of suspension sag, rover tilt, rocker bogie angles, etc. The +X axis points to the front of the rover, +Y to the right side, and +Z down (perpendicular to the chassis deck). See Figure 20-1.



**Figure 20-1: Rover navigation coordinate frame (RNAV)**

The Rover Nav frame is specified via an offset from the current Site frame, and a quaternion that represents the rotation between the two. A new instance of the Rover Nav frame, with a potentially unique offset/quaternion, is created every time the ROVER\_MOTION\_COUNTER increments.

Orientation of the rover (and thus Rover Nav) with respect to Local Level or Site is also sometimes described by Euler angles as shown in Figure 20-2, where  $\psi$  is heading or yaw,  $\theta$  is attitude or pitch, and  $\phi$  is bank or roll.



**Figure 20-2: Yaw, Pitch and Roll Definition**

## **Rover Mechanical (Rover Mech) Frame**

The Rover Mechanical (RMECH) frame is oriented identically to the Rover Nav frame. The origin is forward of Rover Nav by  $x=0.09002$  meters with a Z origin at the top of the deck. Given a point expressed in Rover Mech, if you add  $(0.09002, 0.0, -1.13338)$  you will get the same point expressed in Rover Nav. Rover Mech is not used by any nominal products but could appear in certain special products, generally having to do with arm kinematics.

## **Local Level Frame**

The Local Level frame is coincident with the Rover Nav frame, i.e. they share the same origin at all times. The orientation is different, however. The +X axis points North, +Z points down to nadir along the local gravity vector, and +Y completes the right-handed system. Thus the orientation matches the orientation of Site frames.

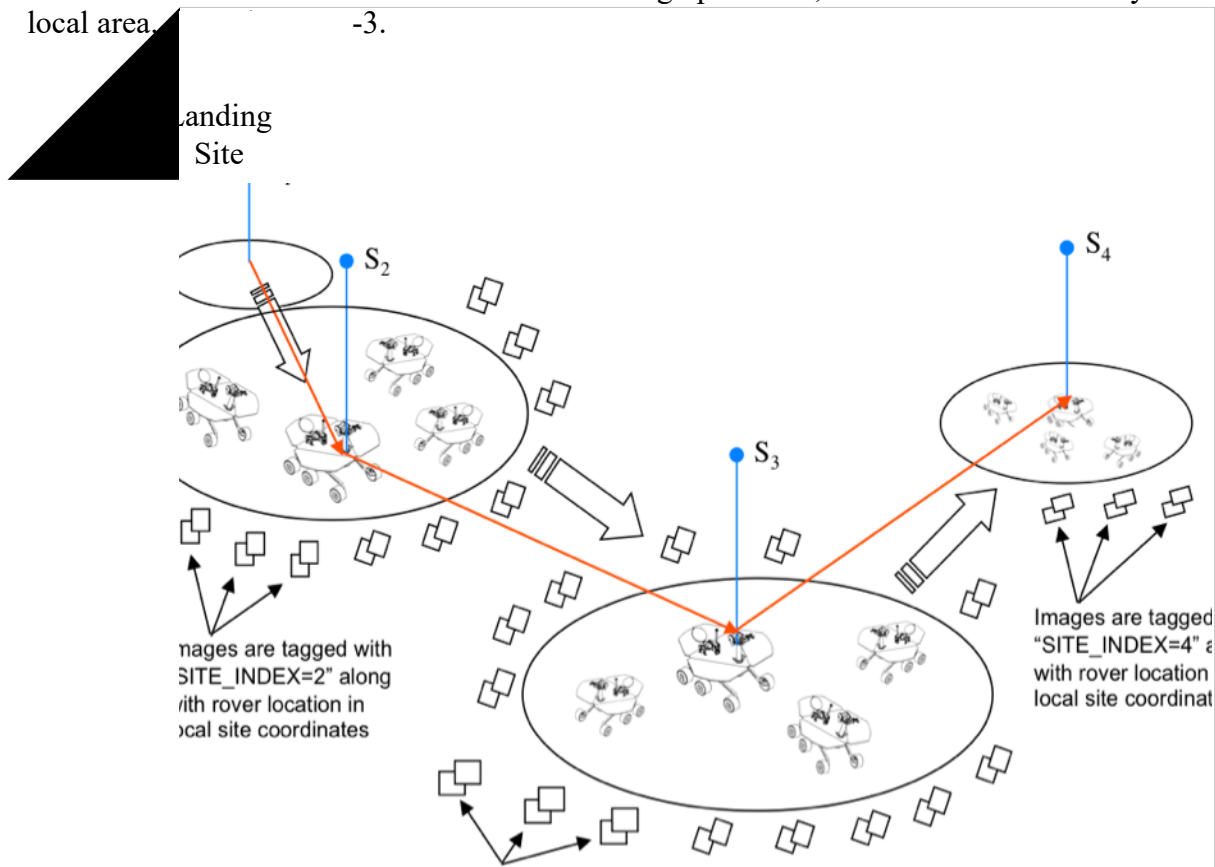
Local Level frames are defined by an offset from the current Site frame, with an identity quaternion.

## **Site Frame**

Site frames are used to reduce accumulation of rover localization error. They are used to provide a common reference point for all operations within a local area. Rover Nav and Local Level frames are specified using an offset from this origin. When a new Site is declared, that becomes the new reference, and the offset is zeroed. In this way, long-term localization error is relegated to the offset between Sites, becoming irrelevant to local operations, because the positions are reset with each new Site.

When a Site frame is declared, it is identical to the Local Level frame, sharing both orientation and position. However, the Site frame is fixed to the Mars surface; when the rover moves, Local Level moves with it but Site stays put. Therefore, for the Site frame, +X points North, +Z points down to nadir along the local gravity vector, and +Y completes the right-handed system.

Sites are indexed, meaning there are multiple instances. Site 1 by definition represents the landing location. New Sites are declared as needed during operations, as the rover moves away from the local area.



**Figure 20-3: Site and Rover frame examples**

The PLACES database (see Section 15.3) stores the set of all site-to-site offsets; such offsets are not in every image label.

## RSM Frame

The RSM frame is attached to the Remote Sensing Mast (RSM) camera head, and moves with it. See the 3PCS[FDD\_PPPCS] for specific definition. It is expressed as an offset and quaternion from the Rover Nav frame.

## Arm Frames

The frame representing the currently selected arm tool is reported in the arm coordinate system group. The selected tool, given by ARTICULATION\_DEV\_INSTRUMENT\_ID, is arbitrary for

any given image and may be surprising; for example SHERLOC-WATSON may not be the selected tool for a SHERLOC-WATSON image. The various tool frames are attached to and aligned with the tool in some manner specific to that tool. See [FDD\_PPPCS] for actual frame definitions.

## 20.8 PIXL Frames

Because PIXL has a moving hexapod as part of the instrument, it has additional coordinate frames. See [FDD\_RAPP], also [Allwood2020].

### 20.8.1 PIXL\_BASE frame

The PIXL Base frame is fixed relative to the rover arm turret. The base coordinate frame origin is located at the junction of the mounting surface and two edges of the mechanical base frame that is bolted to the turret. +Z points in the direction of observation when PIXL is at its home position, +Y points away from the turret joint, and +X completes the right-hand frame. All onboard calculations and pointing information are done in base frame.

### 20.8.2 PIXL\_TOOL frame

The PIXL Tool frame origin is centered on the X-Ray tube output location, at the surface of the instrument when in the home position. +X points out from the instrument, +Y is perpendicular to the X-ray boresight in the direction away from the drill bits, and +Z points toward the turret joint.

### 20.8.3 PIXL\_SENSOR frame

The PIXL sensor head moves with the base coordinate frame to move the X-ray focal point to a desired location by changing the lengths of the struts. The sensor coordinate frame is fixed to the sensor head and moves with it. The orientation of the sensor frame is that same as the base frame when PIXL is in the home position. PIXL MCC camera models are expressed in the sensor frame.

### 20.8.4 PIXL\_CAMERA frame

The camera coordinate frame is also fixed within the sensor head, but is offset from the sensor frame so that its origin is located at the effective camera pinhole. It is also tilted by 18° to align its Z axis with the camera boresight. The camera frame is used for internal geometric calculations but does not appear in image labels or the ground system.



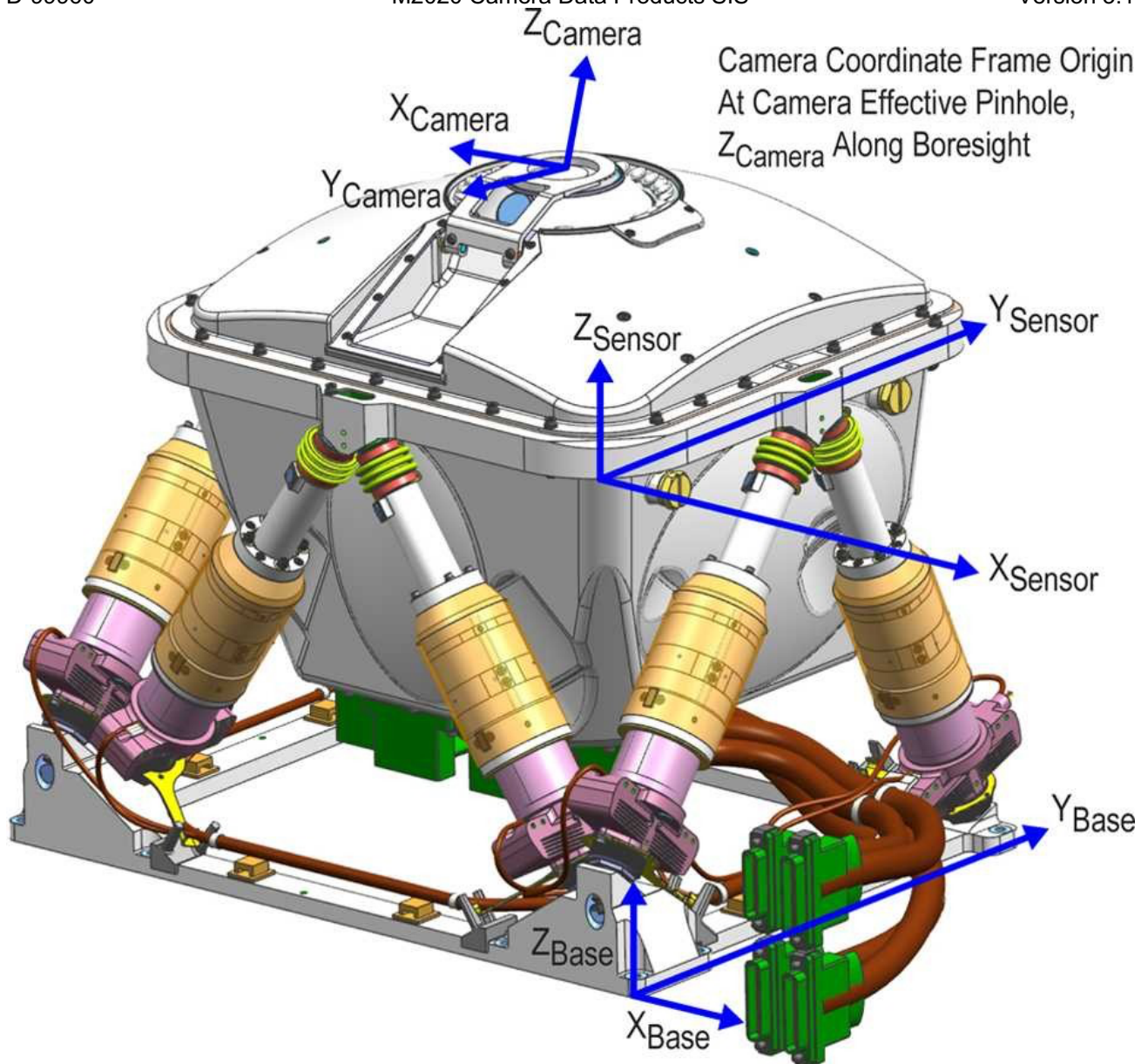
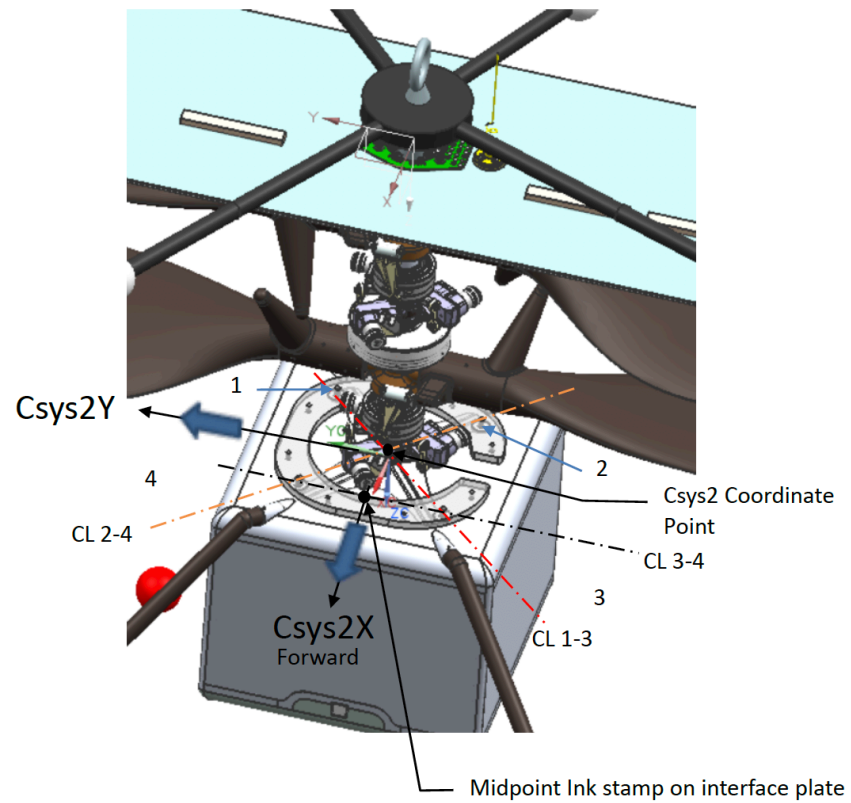


Figure 20-4: PIXL Coordinate Frames

## 20.9 Helicopter Frames

Helicopter-relative frames are fixed to the helicopter body translate and rotate with the helicopter, and maintain a fixed relationship with each other.



**Figure 20-5: Helicopter M(aka Csys2) Frame**



### 20.9.1 Helicopter Mechanical Frame (“M” or “Csys2”)

This is the main body-fixed frame used to reference the placement and orientation of components on the vehicle (see Figure 20-5). The direction of the x axis defines what is referred to as the forward direction on the vehicle; the y axis defines the rightward direction on the vehicle; and the z axis defines the downward direction on the vehicle.

The origin of the frame is defined as being located at the intersection of the plane formed by the top surface of the leg plate inserts, and the vertical line through the center of the mast. The x axis is defined as pointing from the origin through the halfway point of the line segment between the center on the top surface of leg plate inserts 3 and 4. The z axis is defined as pointing downward along the center of the mast (from the rotor toward the legs). The y axis completes the right-handed coordinate system.

The helicopter M frame is analogous to the Rover Nav frame, in that it’s rigidly attached to the vehicle and moves with it.

### 20.9.2 Helicopter IMU Frame (“S1” and “S2”)

These frames define the location and orientation of the sensing axes of the IMU. The frames are defined via IMU specifications. The notation S1 and S2 is used as necessary to distinguish between IMU1 and IMU2. In general, only one will appear in any given helicopter label.

### 20.9.3 Helicopter Ground Frame (“G”)

The ground frame is the frame used by the onboard helicopter software for navigation. For helicopter operations on Mars, the frame is defined on each helicopter GNC flight software startup to coincide with the initial helicopter location and orientation. Waypoints are specified in the Ground Frame, and the relationship to the Rover Site Frame must be manually managed on a flight-by-flight basis on the ground.

The frame is defined such that the z axis points downward along the plumb-bob gravity vector. For helicopter operations on Mars, the x axis is defined as the projection of the M frame x axis onto the horizontal plane (perpendicular to the G Frame z axis) prior to flight. The origin is defined as the intersection of the ground with the vertical line passing through the origin of the Helicopter CM (H) Frame.

The helicopter G frame is analogous to the Site frame, in that it’s rigidly attached to the ground and positions and orientation (M frames) within a local area (flight) are defined relative to the G frame. G frame is not oriented north, however.

## 20.10 Lander Vision System Camera Frames

### 20.10.1 Mars Centered Mars Fixed Frame (“MCMF”)

- **Mars (Planet) -Centered Mars Fixed (P)**
  - Origin: Center of Mars
  - Orientation:
    - > X = lies along the intersection of the Martian reference meridian and the Martian reference ellipsoid equator (“Prime Meridian” defined by the crater Airy-0)
    - > Z = lies parallel to the Mars rotation axis
    - > Y = completes the right-handed set

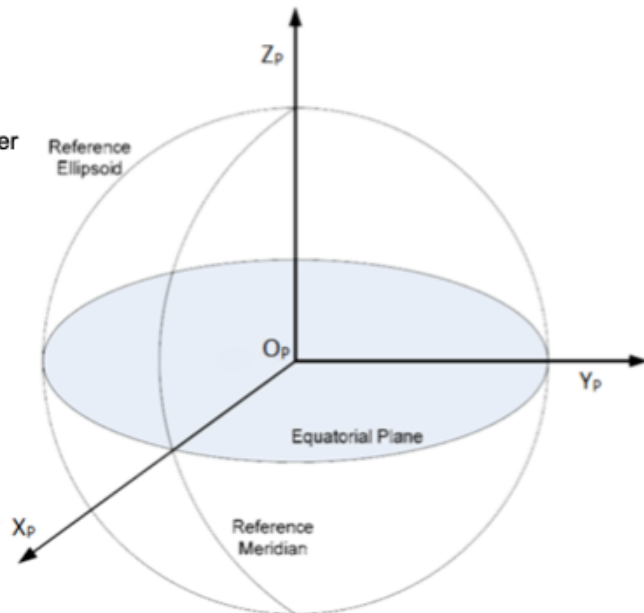


Figure 20-6: Mars Centered Mars Fixed

### 20.10.2 LCAM Internal Frame (“Cint”)

- **LVS Camera internal frame (Cint)**
  - Origin: projection center of the camera (“C” in camera CAHVORE model)
  - Orientation:
    - >  $X_{Cint}$  = Complete right-hand system (notionally along LCAM rows)
    - >  $Y_{Cint}$  = Along LCAM V’ vector (columns)
    - >  $Z_{Cint}$  = Along LCAM A vector (boresight)

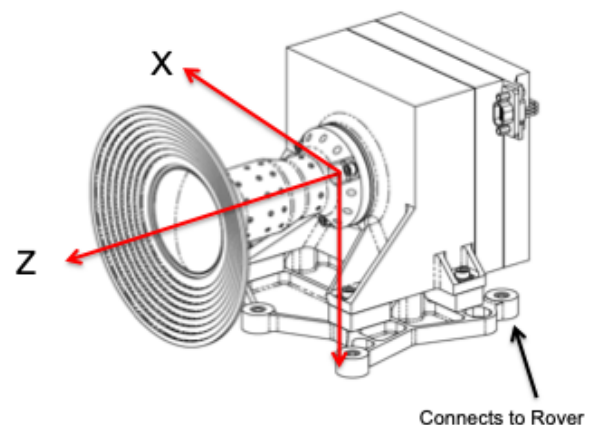
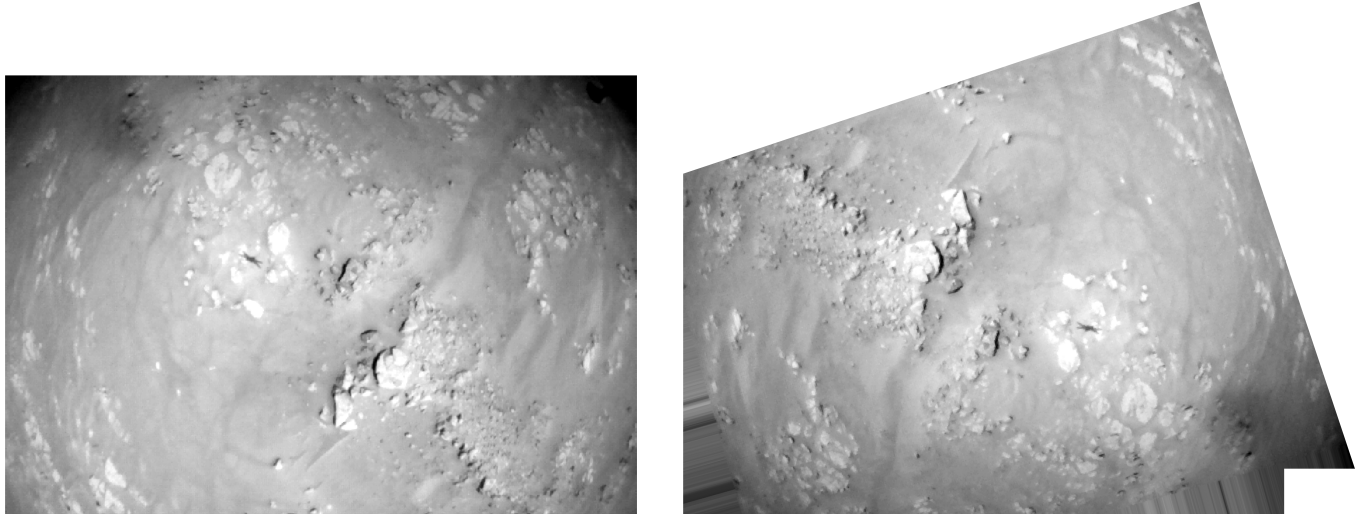


Figure 20-7: LCAM internal frame (Cint)

This document has been reviewed and determined not to contain export controlled technical data.

## Example Coordinate Frame Transform

This section works through a concrete example of how to do coordinate frame transforms using the label items.



**Figure 20-8: Sample helicopter image from Flight 12**  
(HNM\_0174\_0682390538\_578FDR\_N0120001HELI01649\_0000LUJ01.IMG).

Scenario: we are interested in the Helicopter Navcam image shown in Figure 20-8. Where is it on the planet? The left side is the raw image; the right has been rotated 164.2 degrees (explained at the end of this section).

In this section we will refer to the ODL/VICAR labels, but the same information is available in the PDS4 labels.

The first step is to figure out where the image is within the flight. The Heli M frame is attached to the helicopter, while the Heli G frame is the takeoff point. We look at the HELI\_M\_COORDINATE\_SYSTEM definition:

```
GROUP                                = HELI_M_COORDINATE_SYSTEM
  COORDINATE_SYSTEM_NAME              = "HELI_M_FRAME"
  SOLUTION_ID                        = "TELEMETRY"
  COORDINATE_SYSTEM_INDEX             = (12,1649)
  COORDINATE_SYSTEM_INDEX_NAME        = ("FLIGHT","POS")
  ORIGIN_OFFSET_VECTOR                = (-31.662721,77.526172,-10.522429)
  ORIGIN_ROTATION_QUATERNION          = (0.978486,-0.04079,-0.039644,0.198319)
  POSITIVE_AZIMUTH_DIRECTION          = CLOCKWISE
  POSITIVE_ELEVATION_DIRECTION        = UP
  REFERENCE_COORD_SYSTEM_NAME         = "HELI_G_FRAME"
  REFERENCE_COORD_SYSTEM_INDEX        = 12
END_GROUP                            = HELI_M_COORDINATE_SYSTEM
```

That same definition in the PDS4 label looks like this:

This document has been reviewed and determined not to contain export controlled technical data.

```

<geom:Coordinate_Space_Definition>
  <local_identifier>HELI_M_FRAME_12_1649</local_identifier>
  <local_identifier>HELI_M_FRAME_12_1649_TELEMETRY</local_identifier>
  <geom:positive_azimuth_direction>Clockwise</geom:positive_azimuth_direction>
  <geom:positive_elevation_direction>Up</geom:positive_elevation_direction>
  <geom:Coordinate_Space_Present>
    <geom:Coordinate_Space_Indexed>
      <geom:coordinate_space_frame_type>HELI_M_FRAME</geom:coordinate_space_frame_type>
      <geom:Coordinate_Space_Index>
        <geom:index_id>FLIGHT</geom:index_id>
        <geom:index_value_number>12</geom:index_value_number>
      </geom:Coordinate_Space_Index>
      <geom:Coordinate_Space_Index>
        <geom:index_id>POS</geom:index_id>
        <geom:index_value_number>1649</geom:index_value_number>
      </geom:Coordinate_Space_Index>
      <geom:solution_id>TELEMETRY</geom:solution_id>
    </geom:Coordinate_Space_Indexed>
  </geom:Coordinate_Space_Present>
  <geom:Vector_Origin_Offset>
    <geom:x_position unit="m">-31.662721</geom:x_position>
    <geom:y_position unit="m">77.526172</geom:y_position>
    <geom:z_position unit="m">-10.522429</geom:z_position>
  </geom:Vector_Origin_Offset>
  <geom:Quaternion_Plus_Direction>
    <geom:qcos>0.978486</geom:qcos>
    <geom:qsin1>-0.04079</geom:qsin1>
    <geom:qsin2>-0.039644</geom:qsin2>
    <geom:qsin3>0.198319</geom:qsin3>
    <geom:rotation_direction>Forward</geom:rotation_direction>
  </geom:Quaternion_Plus_Direction>
  <geom:Coordinate_Space_Reference>
    <geom:Coordinate_Space_Indexed>
      <geom:coordinate_space_frame_type>HELI_G_FRAME</geom:coordinate_space_frame_type>
      <geom:Coordinate_Space_Index>
        <geom:index_id>SITE</geom:index_id>
        <geom:index_value_number>12</geom:index_value_number>
      </geom:Coordinate_Space_Index>
    </geom:Coordinate_Space_Indexed>
  </geom:Coordinate_Space_Reference>
</geom:Coordinate_Space_Definition>

```

The labels both say that the HELI\_M(12,1649) frame is defined in terms of the HELI\_G(12) frame, with the given offset and quaternion. Note that quaternions are always scalar-first in image labels.

To get the location of the helicopter as expressed in the G frame, we take a point expressed in M frame (the origin, i.e. where the helicopter is), and transform it to G frame using the general frame transform:

$$P_A = Q_{BA} \circ P_B + O_{BA}$$

where  $P_B$  is a point in B frame,  $P_A$  is the equivalent point in A frame,  $O_{BA}$  is the origin of the B frame expressed in A (the offset from the label), and  $Q_{BA}$  is the quaternion from the label expressing the rotation between B and A frames. Rotation of a vector using a quaternion is represented in the section using the notation:

$$V' = Q \circ V$$

and is defined as:

$$V' = Q * (0, V) * (-Q)$$

where  $-Q$  is the inverse quaternion (negate all the vector components but not the scalar), and  $(0, V)$  is a quaternion built with the given vector and 0 for the scalar component.

Quaternion multiplication is defined as:

$$Q = (s, v)$$

$$Q_1 * Q_2 = ((s_1 s_2 - v_1 \cdot v_2), (s_1 v_2 + s_2 v_1 + v_1 \times v_2))$$

Where  $\cdot$  is the vector dot product and  $\times$  is the vector cross product.

So for this particular example we have:

$$P_M = (0,0,0)$$

$$Q_{MG} = (0.978486, -0.04079, -0.039644, 0.198319)$$

$$O_{MG} = (-31.662721, 77.526172, -10.522429)$$

$$P_G = Q_{MG} \circ P_M + O_{MG}$$

$$P_G = (-31.662721, 77.526172, -10.522429)$$

Thus  $P_G$  is the position of the helicopter as measured in G frame. The rotation does not matter here since we're looking at the origin. Looking again at the label, we find the definition of the G frame:

```
GROUP                                = HELI_G_COORDINATE_SYSTEM
COORDINATE_SYSTEM_NAME              = "HELI_G_FRAME"
SOLUTION_ID                         = "TELEMETRY"
COORDINATE_SYSTEM_INDEX             = 12
COORDINATE_SYSTEM_INDEX_NAME        = "FLIGHT"
ORIGIN_OFFSET_VECTOR                = (-709.24329, -680.69713, -0.052)
ORIGIN_ROTATION_QUATERNION          = (-0.9428806983295449, -0.0, 0.0,
0.3331305880845975)
POSITIVE_AZIMUTH_DIRECTION           = CLOCKWISE
POSITIVE_ELEVATION_DIRECTION         = UP
REFERENCE_COORD_SYSTEM_NAME          = "SITE_FRAME"
REFERENCE_COORD_SYSTEM_INDEX         = 3
END_GROUP                           = HELI_G_COORDINATE_SYSTEM
```

Which says that the HELI\_G(12) frame is defined in terms of the rover's SITE(3) frame (which happens to be the landing site for the rover). Applying the same transform as above, we have:

$$P_G = (-31.662721, 77.526172, -10.522429)$$

$$Q_{GS} = (-0.942881, 0, 0, 0.333131)$$

$$O_{GS} = (-709.24329, -680.69713, -0.052)$$

$$P_S = Q_{GS} \circ P_G + O_{GS}$$

$$P_S = (-685.176, -600.487, -10.574)$$

Note that HELI\_S1 and HELI\_S2 frames are IMU frames attached to the helicopter (Section 20.9.2) and should not generally be used. Do not confuse them with the rover SITE frame.

To find that location on the base map, we turn to the Rover PLACES bundle in PDS. This part will change per delivery, as the base map is adjusted every delivery. The numbers below are for Release 3. The best\_tactical.csv file contains the location of the rover at each localization point:

[https://pds-geosciences.wustl.edu/m2020/urn-nasa-pds-mars2020\\_rover\\_places/data\\_localizations/best\\_tactical.csv](https://pds-geosciences.wustl.edu/m2020/urn-nasa-pds-mars2020_rover_places/data_localizations/best_tactical.csv)

```
frame,site,drive,pose,landing_x,landing_y,landing_z,northing,easting,planetocentric_latitude,planetod
etic_latitude,longitude,elevation,map_pixel_line,map_pixel_sample,dem_pixel_line,dem_pixel_sample,rol
l,pitch,yaw,quat_s,quat_v1,quat_v2,quat_v3,areoid,radius,sclk,sol
...
SITE,3,-1,-1,0.000,0.000,0.000,1093299.695,4354494.086,18.444627146,18.346169209,77.450885729,-2569.9
10,2161.22,4296.34,540.30,1074.09,0.000,-0.000,0.000,1.000000000,0.000000000,0.000000000,0.000000000,
3394507.076,3391937.166,667754529,9
```

This document has been reviewed and determined not to contain export controlled technical data.

The “SITE,3” entry defines the rover position at Site 3. The northing, easting, and elevation values can be added to the  $P_S$  coordinate to get the global position (note: elevation is +Z up while the  $P_S$  coordinate is +Z down):

$$\begin{aligned} \text{northing} &= 1093299.695 + (-685.176) = 1092614.52 \\ \text{easting} &= 4354494.086 + (-600.487) = 4353893.60 \end{aligned}$$

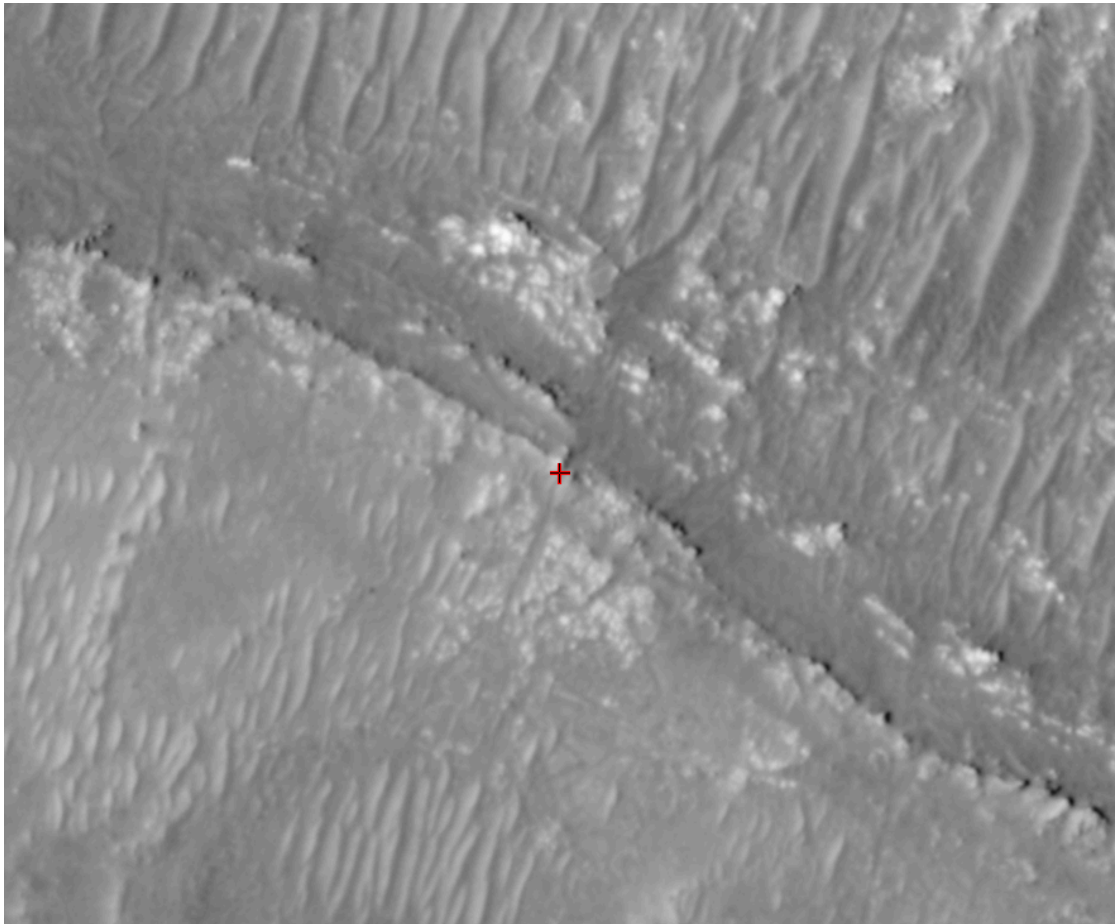
These can then be converted into (planetocentric) latitude/longitude via the formulas in the Rover PLACES SIS:

$$\begin{aligned} \text{lat} &= \frac{\text{northing}}{R_e} \times \frac{180}{\pi} = 18.4330668171 \\ \text{lon} &= \frac{\text{easting}}{R_e \cos(\phi_1(\pi/180))} \times \frac{180}{\pi} = 77.4402019369 \end{aligned}$$

where  $R_e$  is the ellipsoid radius (3396190 m) and  $\phi_1$  is the standard parallel (18.4663 degrees).

The PLACES delivery provides a cutout of the orbital base map covering the traverse. The pixel location of the landing site is given in the csv file above, as `map_pixel_line` and `map_pixel_sample`. Since the map is 25cm/pixel (4 pixels per meter), the  $P_S$  location can be determined from the above:

$$\begin{aligned} \text{line} &= \text{map\_pixel\_line} - x * 4 = 2161.22 + 685.176 * 4 = 4901.92 \\ \text{samp} &= \text{map\_pixel\_samp} + y * 4 = 4296.34 - 600.487 * 4 = 1894.39 \end{aligned}$$



**Figure 20-9: Orbital map of helicopter location (cutout from base map)**

Referring to the map delivered with PLACES:

[https://pds-geosciences.wustl.edu/m2020/urn-nasa-pds-mars2020\\_rover\\_places/data\\_maps/m20\\_orbital\\_map.img](https://pds-geosciences.wustl.edu/m2020/urn-nasa-pds-mars2020_rover_places/data_maps/m20_orbital_map.img)

That coordinate is marked with a red + in Figure 20-9. Comparison with the right side of Figure 20-8 shows they are indeed the same location.

Elevation is derived using the DEM:

[https://pds-geosciences.wustl.edu/m2020/urn-nasa-pds-mars2020\\_rover\\_places/data\\_maps/m20\\_orbital\\_dem.img](https://pds-geosciences.wustl.edu/m2020/urn-nasa-pds-mars2020_rover_places/data_maps/m20_orbital_dem.img)

Using a similar calculation to above, except that we use dem\_pixel\_line and dem\_pixel\_sample, and the DEM resolution is 1m/pixel:

$$\begin{aligned} \text{line} &= \text{dem\_pixel\_line} - x * 1 = 540.30 + 685.176 = 1225.48 \\ \text{samp} &= \text{dem\_pixel\_samp} + y * 1 = 1074.09 - 600.487 = 473.603 \end{aligned}$$

The DEM at that coordinate has a value of -2569.58 meters (subpixel interpolation could be used).

The helicopter's Z coordinate in Site 3 frame (from  $P_3$ ) is -10.574. The elevation of Site 3 (from the PLACES csv file) is -2569.910. This is where we must be careful: "elevation" is defined as positive up, while "Z" is defined as positive down. So to get the helicopter elevation with respect to the Mars datum, we must invert the Z coordinate when combining them:  $-2569.910 + 10.574 = -2559.34$ .

The helicopter's elevation above the ground at this point in its flight is thus  $-2559.34 - (-2569.58) = 10.24$  meters.

Finally, let us consider orientation. The orientation of the helicopter is expressed by the  $Q_{MG}$  quaternion, measured in G frame. If we want to express this in Site 3 frame (in order to get a north/east/down reference), we simply composite the two quaternions:

$$\begin{aligned} Q_{MS} &= Q_{GS} * Q_{MG} \\ Q_{MS} &= (-0.988662, 0.051667, 0.023791, 0.138973) \end{aligned}$$

To convert this to roll/pitch/yaw, the quaternion is first converted to a rotation matrix:

$$\begin{aligned} Q &= (s, x, y, z) \\ R &= \begin{bmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} s^2 + x^2 - y^2 - z^2 & 2xy - 2sz & 2xz + 2sy \\ 2xy + 2sz & s^2 - x^2 + y^2 - z^2 & 2yz - 2sx \\ 2xz - 2sy & 2yz + 2sx & s^2 - x^2 - y^2 + z^2 \end{bmatrix} \end{aligned}$$

which can then be decomposed into the Euler angles:

$$\begin{aligned} \text{roll} &= \tan^{-1}(R_{21}, R_{22}) \\ \text{pitch} &= \tan^{-1}\left((-R_{20}), \sqrt{R_{00}^2 + R_{10}^2}\right) \\ \text{yaw} &= \tan^{-1}(R_{10}, R_{00}) \end{aligned}$$

For this example, we get:

$$\begin{aligned} \text{roll} &= -5.49335 \text{ degrees} \\ \text{pitch} &= -3.52035 \text{ degrees} \\ \text{yaw} &= -15.83397 \text{ degrees} \end{aligned}$$

The cameras are on the +y side of the vehicle as measured in the HELI\_M frame, with the RTE looking in the direction of the +y axis (but 45 degrees down). For the Heli Navcams, the +y axis is to the left of the image, with the +x axis to the bottom of the image. They thus require a 180 degree rotation to put +x at the top of the image. Since the yaw is -15.8 degrees (and is measured from +x), that requires an overall 164.2 degree clockwise rotation of the image to align it with the standard north=up for maps. This rotation is indicated in the right site of Figure 20-8, and matches the orientation of the map in Figure 20-9.

Although this example was for the helicopter, the same transforms apply to the rover. The Rover Nav frame is the general equivalent of Heli M, and Site frame is the general equivalent of Heli G. Site to Site transforms are easier than G to Site because all Site frames are oriented the same. The Rover Nav frame quaternion can be decomposed into roll, pitch, and yaw using the formula above, and points or vectors expressed in any frame can be transformed into any other using the general frame transform formula and the coordinate system transforms from the label. Note that if you want to go “backwards”, from the reference frame to the current frame, you simply take the general formula:

$$P_A = Q_{BA} \circ P_B + O_{BA}$$

and invert it:

$$P_B = -Q_{BA} \circ (P_A - O_{BA})$$



## 21. APPLICABLE SOFTWARE

The instrument data downlink processing software is focused on rapid reduction, calibration, and visualization (in the case of images) of products in order to make discoveries, to accurately and expeditiously characterize the geologic environment around the rover, and to provide timely input for operational decisions concerning rover navigation and Arm target selection. Key software tools have been developed at JPL as part of the IDS and RPS subsystems, and at LANL/IRAP/CNES by the SuperCam team. These toolsets can be used to process data to yield substantial scientific potential in addition to their operational importance.

### 21.1 Utility Programs

Table 7.1 lists (in no particular order) the primary software tools that will be used to process and manipulate downlinked M2020 instrument payload data. At JPL, the “M2020edrgen” program will generate EDRs and the Mars Program Suite of VICAR programs will generate RDRs in PDS format. An IDS pipeline system will deliver the products to the FEI server for transfer to M2020’s ODS as rapidly as possible after receipt of telemetry.

**Table 21-1 – Key software toolsets**

Name	Description	Primary Development Responsibility
SuperCam Ops Software	LIBS software development is primarily performed in IDL code, with LANL (New Mexico) as lead institute.	Dot Delapp (LANL)
	RMI software development is primarily performed in IDL code, with IRAP (France) as lead institute.	Olivier Gasnault (IRAP)
M2020edrgen	Fetches the image Data Product Object (DPO) records from M2020 Data Product (DP) files, reconstructing the image file from the telemetry data into a PDS-labelled image EDR data product. VICAR code.	Alice Stanboli, Hyun Lee (JPL / MIPL)
IDS/Marsviewer	Desktop and Web-based Applications for viewing EDR/FDR/RDR Imaging Products.	Stirling Algermissen (JPL) Nicholas Toole (JPL)
IDS/DataDrive	An interface to search for and download data products in the mission data store.	Jeff Liu (JPL)
IDS/Camp	A web-based geographical information system (GIS) that provides orbital context and enables science team members to spatially capture interest and science intent for strategic operations	Fred Calef (JPL)
Mars Program Suite	Stereo image processing software using EDRs or calibrated images (RDRs), image mosaicking software, 3-D terrain building software. VICAR code: <ul style="list-style-type: none"> <li>MARSINVERTER – Generates inverse lookup table (ILUT) products.</li> <li>MARSDEBAYER – Generates de-Bayered images.</li> <li>MARSCAHV – Generates a geometrically corrected version of the EDR, applying the C, A, H and V camera model vectors.</li> <li>MARSRAD – Generates a radiometrically corrected image from a single input EDR.</li> <li>MARSECORR – Generates a disparity map from a stereo pair of input EDRs, using on-the-fly epipolar alignment.</li> <li>MARSCOR3 – Refines a disparity map from a stereo pair of input EDRs, applying a 2-D correlator.</li> <li>MARSXYZ – Generates an XYZ image from an input disparity map.</li> <li>MARSRFILT – Range filtering for XYZ images.</li> </ul>	Bob Deen (JPL / MIPL)

This document has been reviewed and determined not to contain export controlled technical data.

Name	Description	Primary Development Responsibility
	<ul style="list-style-type: none"> <li>• MARS RANGE – Generates a range image from an input XYZ map.</li> <li>• M2020 REACH – Generates an arm reachability map from an input XYZ map.</li> <li>• M2020 ROUGH – Generates roughness maps.</li> <li>• MARS ERROR – Generates XYZ and range error maps.</li> <li>• MARS SLOPE – Generates slope maps.</li> <li>• MARS BRT – Generates brightness/contrast correction file for mosaic processing.</li> <li>• M2020 FILTER – Generates XML file for image mask files.</li> <li>• MARS FILTER – Generates image mask files.</li> <li>• MARS MASK – Applies image mask files to image files.</li> <li>• MARS DISCOMPARE – Checks consistency for left-to-right and right-to-left stereo image correlations.</li> <li>• MARS UVW – Generates a surface normal image, wherein XYZ is computed normal to the surface.</li> <li>• MARS MAP – Generates a Cylindrical, Polar or Vertical projection mosaic from a list of input EDRs.</li> <li>• MARS MOS – Produces pinhole camera mosaics using uncorrected input images and CAHVOR camera model.</li> <li>• MARS MCAULEY – Generates a combination Cylindrical-Perspective projection mosaic from a list of input EDRs.</li> <li>• MARS TIE – Generates pointing corrections (tiepoint file) from an overlapping set of input EDRs.</li> <li>• MARS NAV – Generates an updated azimuth and elevation file based on comparison with existing image data that can be directly compared.</li> <li>• XVD – De facto image reader software capable of displaying VICAR-labeled image files.</li> </ul>	
	<ul style="list-style-type: none"> <li>• CRUMBS – 3-D terrain building software</li> <li>• MARS MESH – Vicar-based 3D terrain building software based on CRUMBS</li> <li>• MeshQC – Visual Mesh Quality Assessment Tool</li> </ul>	Oleg Pariser (JPL / MIPL)
RPS / RSVP	Visualization, planning, and sequence generation software for use by Sequence Team to create Sol sequences based on activity lists generated by PSI during planning meetings. Java, C and C++ code.	Jeng Yen (JPL)
RPS/ASTTRO	Science Targeting, Planning and Product Visualization Software.	Parker Abercrombie (JPL)

## 21.2 Applicable PDS Software Tools

PDS-labeled images and tables can be viewed with the program PDSView, developed by the PDS and available for a variety of computer platforms from the PDS web site <https://github.com/NASA-PDS/pds-view>. There is no charge for PDSView.

## 21.3 Software Distribution and Update Procedures

The Mars Image Processing Program Suite is available to researchers and academic institutions. Refer to the MIPL Web site at <http://www-mipl.jpl.nasa.gov> for contact information.

## APPENDIX A. DATA PRODUCT AND DATA PRODUCT OBJECT NAMES

Instrument	APID #	APID Names C=Cruise, S=Surface, B=Both	DPOs Used as Sources for Metadata		
			IDPH	Ancillary	Supplementary
Hazcams		<b>Fullframe, Subframe, Downsampled Images:</b>	ImgIdph (B)	n/a	n/a
	317	ImgImageFhl (B)			
	318	ImgImageFhr (B)			
	333	ImgImageRhl (B)			
	334	ImgImageRhr (B)			
		<b>Fullframe, Subframe, Downsampled Images (ICER):</b>			
	319	ImgImageIcerFhl (B)			
	320	ImgImageIcerFhr (B)			
	323	ImgImageIcerRhl (B)			
	324	ImgImageIcerRhr (B)			
		<b>Fullframe, Subframe, Downsampled Images (LOCO):</b>			
	325	ImgImageLocoFhl (B)			
	326	ImgImageLocoFhr (B)			
	329	ImgImageLocoRhl (B)			
	330	ImgImageLocoRhr (B)			
		<b>Fullframe, Subframe, Downsampled Images (JPEG):</b>			
	50	ImgImageJpegFhl (S)			
	51	ImgImageJpegFhr (S)			
	52	ImgImageJpegRhl (S)			
	53	ImgImageJpegRhr (S)			
		<b>Thumbnail Images:</b>	ImgIdph (B)	n/a	n/a
	362	ImgThumbFhl (B)			
	363	ImgThumbFhr (B)			
	378	ImgThumbRhl (B)			
	379	ImgThumbRhr (B)			
		<b>Thumbnail Images (ICER):</b>			
	364	ImgThumbIcerFhl (B)			
	365	ImgThumbIcerFhr (B)			
	368	ImgThumbIcerRhl (B)			
	369	ImgThumbIcerRhr (B)			
		<b>Thumbnail Images (LOCO):</b>			
	370	ImgThumbLocoFhl (B)			
	371	ImgThumbLocoFhr (B)			
	374	ImgThumbLocoRhl (B)			
	375	ImgThumbLocoRhr (B)			
		<b>Thumbnail Images (JPEG):</b>			
	57	ImgThumbJpegFhl (S)			
	58	ImgThumbJpegFhr (S)			
	59	ImgThumbJpegRhl (S)			
	60	ImgThumbJpegRhr (S)			
		<b>Histogram Products:</b>	ImgIdph (B)	n/a	n/a
	311	ImgHistogramFhl (B)			
	312	ImgHistogramFhr (B)			
	315	ImgHistogramRhl (B)			
	316	ImgHistogramRhr (B)			
		<b>Row-summed Products:</b>	ImgIdph (B)	n/a	n/a
	355	ImgRowsumsFhl (B)			
	356	ImgRowsumsFhr (B)			

This document has been reviewed and determined not to contain export controlled technical data.

Instrument	APID #	APID Names C=Cruise, S=Surface, B=Both	DPOs Used as Sources for Metadata		
			IDPH	Ancillary	Supplementary
	359 360	ImgRowsumsRhl (B) ImgRowsumsRhr (B)			
	303 304 307 308	<b>Column-summed Products:</b> ImgColsumsFhl (B) ImgColsumsFhr (B) ImgColsumsRhl (B) ImgColsumsRhr (B)	ImgIdph (B)	n/a	n/a
<b>Navcam</b>	331 332  321 322  327 328  54 55	<b>Fullframe, Subframe, Downsampled Images:</b> ImgImageNl (B) ImgImageNr (B)  <b>Fullframe, Subframe, Downsampled Images (ICER):</b> ImgImageIcerNl (B) ImgImageIcerNr (B)  <b>Fullframe, Subframe, Downsampled Images (LOCO):</b> ImgImageLocoNl (B) ImgImageLocoNr (B)  <b>Fullframe, Subframe, Downsampled Images (JPEG):</b> ImgImageJpegNl (S) ImgImageJpegNr (S)	ImgIdph (B)	n/a	n/a
	376 377  366 367  372 373  47 48	<b>Thumbnail Images:</b> ImgThumbNl (B) ImgThumbNr (B)  <b>Thumbnail Images (ICER):</b> ImgThumbIcerNl (B) ImgThumbIcerNr (B)  <b>Thumbnail Images (LOCO):</b> ImgThumbLocoNl (B) ImgThumbLocoNr (B)  <b>Thumbnail Images (JPEG):</b> ImgThumbJpegNl (S) ImgThumbJpegNr (S)	ImgIdph (B)	n/a	n/a
	313 314	<b>Histogram Products:</b> ImgHistogramNl (B) ImgHistogramNr (B)	ImgIdph (B)	n/a	n/a
	357 358	<b>Row-summed Products:</b> ImgRowsumsNl (B) ImgRowsumsNr (B)	ImgIdph (B)	n/a	n/a
	305 306	<b>Column-summed Products:</b> ImgColsumsNl (B) ImgColsumsNr (B)	ImgIdph (B)	n/a	n/a
<b>CacheCam</b>	13  14  15	<b>Fullframe, Subframe, Downsampled Images:</b> ImgImageCc (B)  <b>Fullframe, Subframe, Downsampled Images (ICER):</b> ImgImageIcerCc (B)  <b>Fullframe, Subframe, Downsampled Images (LOCO):</b> ImgImageLocoCc (B)	ImgIdph (B)	n/a	n/a

This document has been reviewed and determined not to contain export controlled technical data.

Instrument	APID #	APID Names C=Cruise, S=Surface, B=Both	DPOs Used as Sources for Metadata		
			IDPH	Ancillary	Supplementary
	56	<b>Fullframe, Subframe, Downsampled Images (JPEG):</b> ImgImageJpegCc (S)			
	16	<b>Thumbnail Images:</b> ImgThumbCc (B)	ImgIdph (B)	n/a	n/a
	17	<b>Thumbnail Images (ICER):</b> ImgThumbIcerCc (B)			
	18	<b>Thumbnail Images (LOCO):</b> ImgThumbLocoCc (B)			
	49	<b>Thumbnail Images (JPEG):</b> ImgThumbJpegCc (S)			
	22	<b>Histogram Products:</b> ImgHistogramCc (B)	ImgIdph (B)	n/a	n/a
	23	<b>Row-summed Products:</b> ImgRowsumsCc (B)	ImgIdph (B)	n/a	n/a
	24	<b>Column-summed Products:</b> ImgColsumsCc (B)	ImgIdph (B)	n/a	n/a
<b>SuperCam RMI</b>	426 640	<b>Fullframe, Subframe Images:</b> ScamScidata (C) ScamRmiImage (S)	ImfldphSubset6 (S)	ScamRmiHeader (S) ScamAncillaryTemperatures (S) ScamAncillaryArgs (S) ScamAncillaryParams (S) ScamFocalDistance (S)	n/a
	415	<b>Fullframe, Subframe Images (ICER):</b> ScamRmiImageIcer (S)			
	642	<b>Fullframe, Subframe Images (JPEG):</b> ScamRmiImageJpeg (S)			
	643	<b>Zstack:</b> ScamRmiZstackImage (S)	ImfldphSubset6 (S)	ScamRmiHeader (S) ScamRmiRawMetadata (S) ScamAncillaryTemperatures (S) ScamAncillaryArgs (S) ScamAncillaryParams (S) ScamFocalDistance (S)	n/a
	644	<b>Zstack (ICER):</b> ScamRmiZstackImageIcer (S)			
	645	<b>Zstack (JPEG):</b> ScamRmiZstackImageJpeg (S)			
<b>Mastcam-Z</b>	418 419 677 689	<b>Images:</b> MczLScidata (C) MczRScidata (C) MczLImage (B) MczRImage (B)	DpoCidph (C) ImfldphSubset6 (S)	MczImageAncillaryData (C) MmmImageAncillaryData (S)	n/a
	682 694	<b>Thumbnail Images:</b> MczLThumbnail (S) MczRThumbnail (S)			
	684 696	<b>Video:</b> MczLVideo (S) MczRVideo (S)			
	679 691	<b>Recovered Product:</b> MczLRecoveredProduct (S) MczRRecoveredProduct (S)	DpoCidph (C) ImfldphSubset6 (S)	MmmRecoveredProductAncillaryData (S)	n/a
	680 692	<b>Recovered Thumbnail:</b> MczLRecoveredThumbnail (S) MczRRecoveredThumbnail (S)			

This document has been reviewed and determined not to contain export controlled technical data.

Instrument	APID #	APID Names C=Cruise, S=Surface, B=Both	DPOs Used as Sources for Metadata		
			IDPH	Ancillary	Supplementary
	685 697	<b>Zstack:</b> MczLZstack (S) MczRZstack (S)	DpoCidph (C) ImfldphSubset6 (S)	MmmZstackAncillaryData (S)	n/a
	678 690	<b>Range Map:</b> MczLRangemap (S) MczRRangemap (S)	DpoCidph (C) ImfldphSubset6 (S)	MmmRangeMapAncillaryData (S)	n/a
<b>SHERLOC WATSON ACI</b>	425 720	<b>Images:</b> SrcImgScidata (C) SrcImgImage (S)	DpoCidph (C) ImfldphSubset6 (S)	MmmImageAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	721	<b>Thumbnail Images:</b> SrcImgThumbnail (S)	DpoCidph (C) ImfldphSubset6 (S)	MmmImageAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	722	<b>Video:</b> SrcImgVideo (S)	DpoCidph (C) ImfldphSubset6 (S)	MmmVideoAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	723	<b>Zstack:</b> SrcImgZstack (S)	DpoCidph (C) ImfldphSubset6 (S)	MmmZstackAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	724	<b>Range Map:</b> SrcImgRangemap (S)	DpoCidph (C) ImfldphSubset6 (S)	MmmRangeMapAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	725	<b>Recovered Product:</b> SrcImgRecoveredProduct (S)	DpoCidph (C) ImfldphSubset6 (S)	MmmRecoveredProductAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	726	<b>Recovered Thumbnail:</b> SrcImgRecoveredThumbnail (S)	DpoCidph (C) ImfldphSubset6 (S)	MmmRecoveredAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	727	<b>Directory List:</b> SrcImgDirectory	ImfldphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	728	<b>Debug Info:</b> SrcImgDebugInfo	ImfldphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	729	<b>Image Status:</b> SrcImgStatus	ImfldphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	730	<b>BIST Status:</b> SrcImgBistStatus	ImfldphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	731	<b>Util Test:</b> SrcImgUtilTest	ImfldphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	732	<b>Memory Dump:</b> SrcImgDumpMemory (S)	ImfldphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	733	<b>Zstack List:</b> SrcImgZstackList (S)	ImfldphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
<b>PIXL</b>	422 35 36	<b>Images:</b> PixlSciEng (C) PixlTracking (S) PixlSci (S) PixlSciDec (S)	PixlTracking: ImfldphSubset4 (S)  PixlSci/Dec: ImfldphSubset1 (S)	n/a	n/a
<b>MEDA Skycam</b>	420 43 62	<b>Fullframe, Subframe, Downsampled Images:</b> MedaSciData (C) MedaUnprocessedImage (S) MedaPartialImage (S)	ImfldphSubset6 (S)	ImgHdr (S)	n/a
	46	<b>Fullframe, Subframe, Downsampled Images (compressed):</b>			

This document has been reviewed and determined not to contain export controlled technical data.

Instrument	APID #	APID Names C=Cruise, S=Surface, B=Both	DPOs Used as Sources for Metadata		
			IDPH	Ancillary	Supplementary
		MedaCompressedImage (S)			
	45	<b>Thumbnail Images:</b> MedaThumbnailImage (S)	ImfldphSubset6 (S)	ImgHdr (S)	n/a
<b><u>EDL Cam</u></b>  <b>PUCAM1</b> <b>PUCAM2</b> <b>PUCAM3</b> <b>DDCAM</b> <b>RDCAM</b> <b>RUCAM</b>	416	<b>Images:</b> ImgEdlcamScidata (B)	n/a	n/a	n/a
<b>LCAM</b>	630 631 631	<b>Images:</b> VcemgrScidata (B) VcemgrLvsPackets (C) VcemgrVcePackets (S)	n/a	VcemgrMeasurementPacket (C)	n/a
<b>Heli</b>	417 417	<b>Images:</b> HeloScidata (C) LeoSciData (S)	n/a	n/a	n/a

## APPENDIX B. PRODUCT LABEL KEYWORD DEFINITIONS, VALUES, AND SOURCES

As described in the main text, there are three types of label keywords: VICAR, ODL, and PDS4. The VICAR and ODL labels are virtually identical and are referred to here collectively as “VICAR” labels.

This Appendix describes several tables that will be useful for understanding the details of these keywords. All of the tables are in separate files within the document collection, with names as described herein.

### **Operations Keyword Table**

This table lists the VICAR keywords as used by the operations team. This is a hand-generated table containing the VICAR keyword and definition, metadata about the valid values, data types, units, and location, and (in the 3<sup>rd</sup> column) where in the telemetry the data comes from. This last column should not be needed for archive users to understand the data, since the telemetry is not archived, but it is critical for ops users, and may provide hints to archive users if the rest of the information is insufficient for some reason.

Archive users should generally avoid this table in favor of the tables below, since it contains no PDS4 label information, and the tables below should have sufficient information. However, it is being included in case it is of use. The tables can be cross-referenced using the VICAR keyword name.

Also note that the information in this table is somewhat outdated and should not be relied upon as a definitive source. It is included to help users understand conceptually how the labels are defined and provide some indication as to where the information comes from in telemetry. Users should consult the PDS4 Keyword Tables for definitive information.

The filename for this table is:

Mars2020\_Camera\_SIS\_Operations\_Label.pdf

### **PDS4 Keyword Tables**

This set of tables describes the PDS4 keywords (classes and attributes in PDS 4 parlance). They include pointers to the matching VICAR keywords, as well as both the generic (multimission) definition of the keyword, and the specific “nuance” or supplemental information that applies only to Mars 2020.

These tables are created by examining a set of sample labels (incorporating all types of products being created) in order to determine the PDS4 classes and attributes that are actually used by the



products described in this SIS. This list is then cross-referenced against the PDS4 data dictionaries in order to find the definitions, children, valid values, and data types. This list is then augmented with “property maps” that provide the Mars 2020-specific valid values, and the “nuance” definitions.

These tables are thus much more useful for most purposes than looking at the PDS4 data dictionaries directly, because they contain *only* the keywords that are *actually* used.

The first column contains the name of the PDS4 attribute (keyword) or class (container), and the PDS4 dictionary it comes from. Along with that, when applicable, are the VICAR keyword and property name(s) from which the values are derived. The property name is the section of the VICAR label. Not every entry has a VICAR keyword; some entries are merely containers, others contain constants or values that are derived in other ways. Some of the VICAR keywords refer to the class rather than the attribute; for example a VICAR vector keyword will typically refer to the vector’s class rather than the x,y,z attributes individually.

The second column contains the definition. There are two components to many definitions, as alluded to above. The first, which is always present, is the standard PDS4 definition that applies to all missions, from the PDS4 data dictionary. The second (in italics) is a Mars 2020-specific nuance to the definition, providing additional context that applies specifically to Mars 2020.

The third column is broken up into several pieces. The first is the XPath. This gives the “path” of where the item can be found in the label, tracing the hierarchy from the root (often but not always Product\_Observational) down to the item itself. Each level in this hierarchy is a hyperlink, which can be clicked on to go directly to that item’s definition.

Underneath the XPath is a field whose content varies based on the type. For attributes (“keyword”), this contains the valid values, when such are defined either by the PDS4 data dictionary or the Mars 2020-specific property maps. For classes (containers), the valid children are listed. Those that are blue hyperlinks are actually used by Mars 2020; clicking on them will go to that item’s definition. Those that are not blue are defined by the PDS4 data dictionary but are not used by Mars 2020.

Finally, also underneath the XPath field is another column containing (for attributes only) the data type and units. All attributes should have a data type, but only some have units defined.

These label tables are the primary source of information regarding the metadata in the labels. The rest of this document describes things at a high level; the label tables (along with the ops label table, above) define specifically what each label item means.

There are two versions of the table: sorted by PDS4 name, and sorted by VICAR keyword. The tables can be used in either direction. Given a label item you don’t understand, you can look it up in the table (sorted by either PDS or VICAR name, depending on which you’re looking at) to find the definition. In some cases you may need to go up the hierarchy to find a meaningful definition (for example the definition for “x” is not particularly useful, but the parent or grandparent should describe what the full x,y,z value is being used for). Alternatively, given an item in the table, you can find the item in the label by following the XPath – looking down the hierarchy of elements until you find the item. Note that not all keywords are in any given label; the table encompasses image products, browse images, mosaics, meshes, calibration, and documentation files.

The cross-reference between PDS4 name and VICAR keyword can also be useful for comparing values across similar missions (MSL, MER, Phoenix, etc) that use PDS3 (the VICAR and PDS3 keywords are generally the same).

Each of the tables is provided in both HTML and PDF format. The files are:

Mars2020\_Camera\_SIS\_Labels\_sort\_pds.pdf  
Mars2020\_Camera\_SIS\_Labels\_sort\_pds.html  
Mars2020\_Camera\_SIS\_Labels\_sort\_vicar.pdf  
Mars2020\_Camera\_SIS\_Labels\_sort\_vicar.html

## APPENDIX C. CALIBRATION PARAMETERS

This Appendix describes the calibration files that are included with the PDS delivery. Because the format of calibration files is basically arbitrary, they are delivered in the format needed by the Mars program suite that generates the bulk of these products. If and when the Mars program suite is made publicly available, these files will be usable with it as-is. The information in the files is useful outside of the Mars program suite context, as they provide information necessary to reproduce or validate the processing in the archive.

From the PDS point of view, they are all plain text supplemental files or (in some cases) images. Some of the text files are in XML format, but they are treated as plain text for PDS purposes. Because PDS reserves the use of the extension “.xml” for label files, the few types of files that should be named “.xml” are instead named “.xmlx” (this is noted again in the descriptions below). These need to be renamed to “.xml” for use with the Mars program suite.

The PDS labels on these files are minimal. Since they effectively *are* metadata, there is little additional metadata that would be useful in the label. All label files have the same basename (including extension) as the file they point to, but have “\_lbl.xml” appended to the filename. The “\_lbl” clearly distinguishes them from the “.xmlx” files mentioned above. All files with a “\_lbl.xml” extension are PDS4 label files, and can be ignored if using these files with the Mars program suite.

The calibration files are organized into four directories. An overview of each type of file is described below.

Many of the filenames start with a “venue”. For Mars 2020 this venue is “M20”, for the flight hardware. Other testbeds, e.g. “M20VSTB”, exist for various testbeds at JPL. Cal data files specific to the testbeds are not included in the delivery, but files that contain information for both testbed and flight retain the testbed values. The inclusion of the mission name allows the multimission Mars software suite to work with data from any mission without collision among calibration files.

In many cases, there are two copies of the cal files. One is the original file as supplied by the camera calibration engineer (often J. Maki). The other is a copy of the file, with the filename needed for use with the Mars software suite. In the operations system, the “copy” is actually a Linux softlink to the original file. Because PDS does not support softlinks, it is a copy in the archive. These are noted as “original” and “softlink” files in the writeups below.

It should be noted that the basic layout of these files is the same for all users of the Mars suite: MER, MSL, Phoenix, and InSight. The details differ a little, however.

### Top-Level Directory

README.TXT: This file contains the history of changes to the calibration files, as well as current status. It is included mostly for historical interest.

### camera\_models Directory

This document has been reviewed and determined not to contain export controlled technical data.

`M20_SN_xxxx.cahv{or}{e}`: These files contain the calibration camera models for each camera. The calibration camera model is transformed by the inverse of the calibration pose and then again by the actual pose of the arm in order to create the camera model for any given image. These are “softlink” files (see Appendix introduction). The SN (serial number) relates to the `camera_mapping` file (described below). The actual camera models are the C,A,H,V,O,R, and optionally E vectors described in the file, and are the only part of the file that is actually used. The rest of the values are covariance matrices, residuals, and other information about the camera model, that is output by the camera calibration software developed by Todd Litwin at JPL.

`M20_SN_xxxx.interp{t,f,z}`: These files are the camera model interpolation files for temperature (Navcam) or zoom, focus, and temperature (Mastcam-Z). They contain the zoom, focus, or temperature value and the specific camera model filename for that point.

Other `*.cahv{or}{e}`: These are “original” camera model files (see Appendix introduction). The `CAL_nnn` allows for multiple recalibrations.

## flat\_fields Directory

`M20_FLAT_SN_xxxx.IMG`: These are flat field images for each camera. The serial number relates to the `camera_mapping` file. The image is divided by the flat field in order to remove flat field effects. These are float-format images, with an average value near 1.0. These are “softlink” files (see Appendix introduction). The files with “RAWBAYER” in their names have the Bayer pattern encoded in them; they should be used for Bayer-pattern images with the non-“RAWBAYER” files used for color images. Some instruments have a `_F_x` suffix indicating they apply to a given filter. There are other suffixes for special calibration images as defined in Section 7.1.1

`M20_FLAT_SN_xxxx.varflat`: These are files that define a list of flat fields to use when they vary as a function of zoom, focus, or temperature. Similar to the camera model `.interp` files, they contain the zoom, focus, or temperature value and the specific flat field filename for that point. Note that these generally point to “original” flat field files rather than “softlink” ones (see Appendix introduction).

Others: Other `.IMG` or `.VIC` files in this directory are the “original” flat field files (see Appendix introduction).

## ilut Directory

`M20_MCZ_LUTn.txt`: These are the inverse lookup table (ilut) files needed to create the decomanded ILT products (Section 5.2.2) for the Mastcam-Z camera. The format is simple: each line has “input\_DN output\_DN” which converts from 0-255 to 0-4095. The same format is used for all files in this directory.

`M20_SI_LUTn.txt`: These are the inverse lookup table files for the Watson camera, in the same format.

`M20_SC_LUTn.txt`: These are the inverse lookup table files for the ACI camera.

`M20_LUTn.txt`: These are the inverse lookup table files for the engineering cameras.

This document has been reviewed and determined not to contain export controlled technical data.

`M20_LUT2_sols0-13.txt`: This is a special version of `M20_LUT2.txt` that was used for sols 0-13 only.

`M20_COMP_LUTn.txt`: These are the inverse lookup table files for the RMI camera.

`rmi_bitscaling*.txt`: These are the “original” files for RMI (see Appendix introduction)

## **param\_files Directory**

`M20_camera_mapping.xmlx`: This file contains information for each camera on Mars 2020. The information includes names and ID’s, serial number, filter status, camera model type, nominal size, and color status. The serial number is used as a lookup key in several other calibration files. This file needs to be renamed with a “.xml” extension if they are to be used with the Mars software suite.

`M20_hardware.point`: This file contains pointing parameters for the mast- or arm- mounted cameras. The calibration position and quaternion define the pose of the mast or arm when the camera models were created. The calibration model is transformed by the inverse of this before being transformed using the actual arm pose or mast kinematics for a given image. The pointing error values are used by the pointing correction mechanism as the initial perturbation for each parameter. Most of the rest of the values are parameters for the mast kinematics (see [Bell2017] for description, which is for MSL but the concept is the same).

`M20_rad_cal.parms`: This file defines the radiometric calibration coefficients for each camera. The responsivity values are used to convert DN/sec (after dividing by exposure time) to radiance units for radiometrically corrected products; the units are thus  $\text{W/m}^2/\text{nm}/\text{steradian}/(\text{DN}/\text{sec})$ .

`M20_rad_cal_rte_whitebal.parms`: This is a special version of `M20_rad_cal.parms` used for the helicopter RTE camera when the image was white balanced onboard.

`M20_rover_filter.xmlf`: This file defines the mask used to filter all images. It contains only a description for a horizon mask with an elevation of 90 degrees, which effectively disables it.

`M20_filter_exclude.txt`: This file defines a list of “objects” to exclude when making a rover mask.

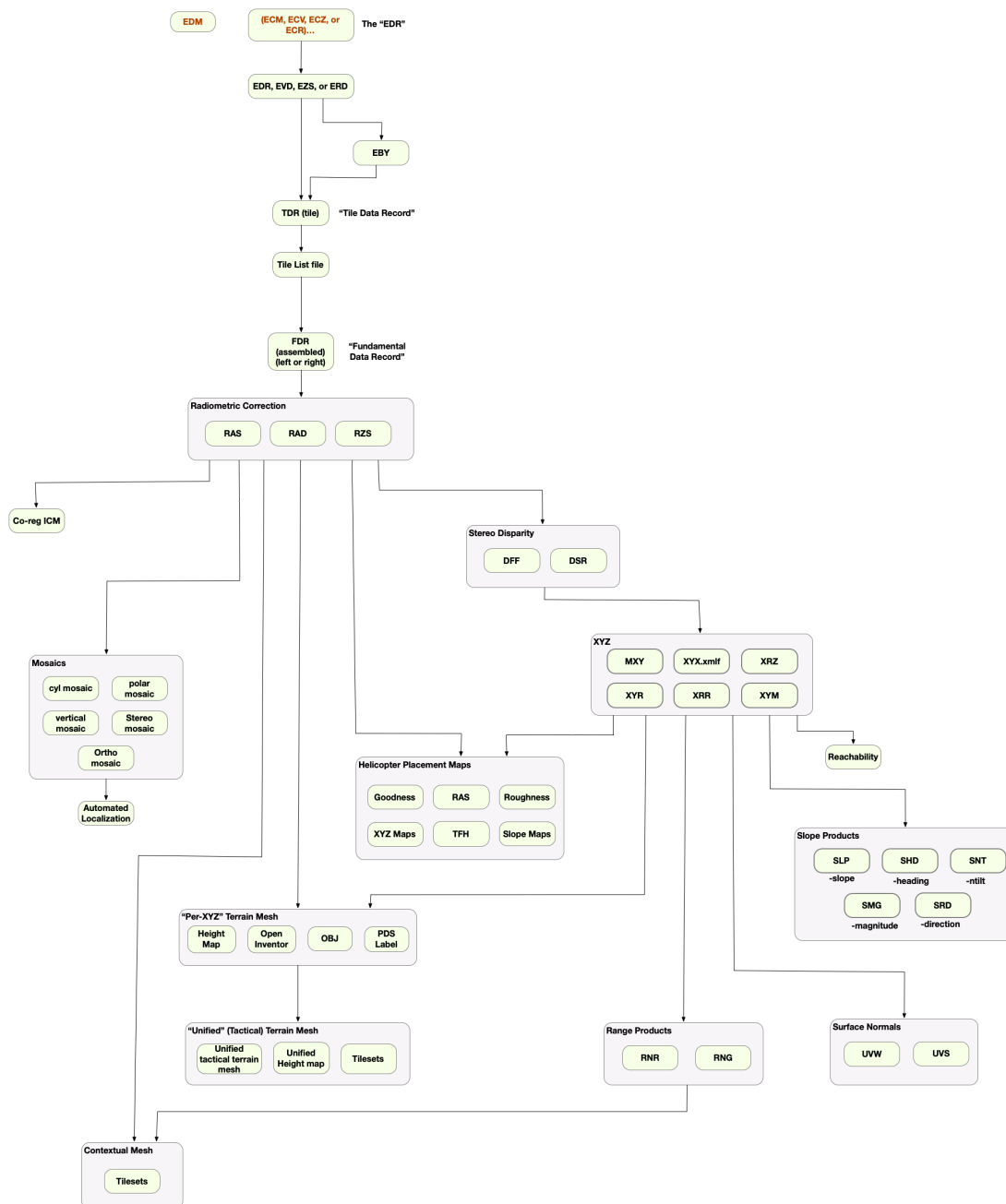
`M20_MCZ_*_zoom_fnum.csv`: This is a CSV file that describes the relationship between zoom motor count (column 1) and f/number (column 2). Column 3 is unused. See Section 7.1.1.3.

`M20_color_cal.parms`: This file contains color calibration parameters for the cameras.

## APPENDIX D. PRODUCT FLOW DIAGRAM

Figure 21-1 shows a high level view of the pipeline data flow. A diagram showing the complete data flow for the pipeline is included as part of this document. For space reasons, it is a separate file, in the same documentation directory/collection in PDS. The file is:

Mars2020\_Camera\_SIS\_Pipeline\_Flow.pdf



**Figure 21-1: Simplified Schematics of core RDR products generation.**

This document has been reviewed and determined not to contain export controlled technical data.

## APPENDIX E. SHERLOC WATSON AND ACI [SUPPLEMENTARY INFORMATION TABLE]

Each Planetary Data System (PDS) release of SHERLOC ACI and WATSON images is accompanied by a document, a table, located in the document camera collection, that provides ancillary information about each image. The table for the first PDS release is:

Mars2020\_SHERLOC\_ACI\_WATSON\_Image\_Information\_Supplement.pdf

The table has 15 columns:

1. sol – the sol that a given image was acquired; the sol is referenced to sol 0, the Martian Sol on which Perseverance landed in Jezero crater.
2. sequence – the SHERLOC command sequence used to acquire the image.
3. camera – the camera used to acquire the image, ACI or WATSON.
4. camera position – a statement about name of the imaging target, the intended distance between the camera and the target, and other information such as whether the image is part of a stereo pair or mosaic.
5. original command downlink bin – information used tactically by the SHERLOC team to understand when a given image might be downlinked from Mars; downlink bins are prioritized from 1 (very high) to 99 (very low).
6. representative EDR file name – the file name of a representative image (or, for onboard focus merge products, a representative image product); text is orange if only a thumbnail image has been received.
7. CDPID (CAMERA\_PRODUCT\_ID) – CAMERA\_PRODUCT\_ID or IMAGE\_ID of each image; these numbers are assigned onboard the instrument and indicate where the image is stored in the shared ACI/WATSON DEA (digital electronics assembly) data storage. Images are numbered consecutively upon acquisition and storage. These numbers are recycled when old data are deleted from the DEA to make room for new data. A combination of Instrument name (WATSON or ACI), sol (of image acquisition), and CDPID uniquely identify each image acquired by the two instruments.
8. SCLK (SPACECRAFT\_CLOCK\_START\_COUNT) – SCLK time reported in the EDR file name, as a reference point to tie this with the filenames of other versions of the same image produced either onboard the instrument (e.g., thumbnails, full images, different data compression) or on Earth (e.g., RDR products).
9. focus type – the commanded method to focus the camera, for a given image, at the motor count value in the following column. Focus types are: auto = autofocus; manual = manual focus; last = focus established in preceding image; relative-z = z-stack relative to preceding focus; manual-z = manual focused z-stack; none = camera was not focused (motor count 0).

This document has been reviewed and determined not to contain export controlled technical data.

10. motor count – stepper motor count (FOCUS\_POSITION\_COUNT) in the INSTRUMENT\_STATE\_RESULTS group reported by the camera for a given image. This indicates camera focus position.
11. range, cm – the working distance (between camera lens and target) or range (for focus stacks in which the working distance remains constant), in centimeters, as estimated from the motor count (FOCUS\_POSITION\_COUNT) using an empirically-derived equation that relates motor count to distance. Currently, the equations used are those reported in [Bhartia 2021], Equation 1 for ACI and Equations 3 and 4 for WATSON. However, it is expected that these equations will change following further assessment of data acquired from Mars. [Bhartia 2021] equation for ACI works well; the equation for WATSON works well over the ~2.1 to ~40 cm range. In some cases, where an “approximate” symbol (~) is presented, the distance or range is an estimate and was not computed from motor count. For focus merge products, the range reported is that of the middle image in the parent focus stack (e.g., in a focus merge of 8 images, it is the range of image 5).
12. range, uncertainty  $\pm$  cm based on DOF – for WATSON, this is the uncertainty in working distance or range determined from motor count (FOCUS\_POSITION\_COUNT), knowledge of the WATSON depth of field (DOF), and the empirically-determined relation between motor count and working distance. Currently, the equation and coefficients used to estimate range uncertainty for WATSON are given in Equation 6 [Bhartia 2021]; the reported value is the average of “near” and “far” depths of field. Note that range uncertainty increases with increasing range. For ACI, uncertainty is minimal and the field is populated with “n/a” for “not available”. It is expected that this information will be refined and equations will be updated following further analysis of data acquired on Mars.
13. pixel scale,  $\mu\text{m}/\text{pixel}$  – pixel scale assumes a flat target parallel to the instrument CCD and is a projection of the pixel onto the target; each square pixel in this case translates to a distance on each side of the pixel which is referred to here as the pixel scale in units of micrometers per pixel. For WATSON, pixel scale is estimated by an empirically derived equation that relates pixel scale to working distance; currently, this field uses Equation 2 [Bhartia 2021]. For ACI, the scale is assumed to be  $10.1 \mu\text{m}/\text{pixel}$  as the small differences based on motor count position have not yet been determined [Bhartia 2021]. It is anticipated that the relevant equation(s) will be changed upon further analysis of data acquired on Mars. For focus merge products, the pixel scale reported is that of the middle image in the parent focus stack (e.g., in a focus merge of 8 images, it is the range of image 5).
14. pixel scale, uncertainty  $\pm \mu\text{m}/\text{pixel}$  – for WATSON, the pixel scale uncertainty is based on the range uncertainty determined using DOF; the reported value is based on the average “near” and “far” depths of field. Note that pixel scale uncertainty increases with increasing range. For ACI, currently no pixel scale uncertainty is reported and the field is reported as “n/a”. It is expected that this information will be refined and equations will be updated following further analysis of data acquired on Mars.
15. image information – image information is a brief description of the image, its target, its intended standoff distance, whether it is part of a focus stack, whether the LEDs were on, whether the dust

This document has been reviewed and determined not to contain export controlled technical data.



cover was closed, whether its purpose was to serve as an autofocus sub-frame, etc. In some cases, post-data-acquisition information is indicated (for example, if the image was unexpectedly out of focus). For focus merge products, the sol and parent CDPIDs (CAMERA\_PRODUCT\_IDs) are indicated so as to help the data user identify the parent images.

## **APPENDIX F. SCLK USAGE IN THE PRODUCTS FILENAMES AND THE METADATA**

As discussed in Section 4.3, most spacecraft events are recorded with spacecraft clock time (SCLK). The following table lists SCLK recorded in the filename of data products or written as a value of various label keywords. It is stored in a separate file in the camera document collection:

Mars20\_SCLK\_usage.pdf

## APPENDIX G. SPECIAL PRODUCT FLAG TABLE

Most products are automatically produced by the IDS pipeline. However, there are a number of products that are produced manually, by analysts during operations. These products are often the most interesting or most highly processed images, because analyst time is limited and is generally focused on those areas where the rover is doing something special or unusual.

All three filename conventions (single-frame in Section 18.1, mosaics in Section 18.2, and meshes in Section 0) contain a character to indicate special processing.

This special processing flag is “\_” to indicate nominal, pipeline-produced products. Any other character indicates some kind of special processing.

Special processing flags are assigned ad-hoc, as needed during operations. A table describes what characters have been assigned and (in general terms) what processing was done. This table is provided in the camera documentation collection:

Mars2020\_Special\_Processing\_Flags.pdf

## APPENDIX H. CAMERA CHARACTERISTICS TABLES

A summary of the basic characteristics of each camera is stored in a separate file in the camera document collection:

Mars2020\_Camera\_Basics.pdf

A more detailed set of camera characteristics is also available in a separate file in the camera document collection:

Mars2020\_Camera\_Characteristics.pdf

These tables are meant to supplement the information provided in Section 2. They are provided as a convenience to the user; the publications listed in the Basics table are the definitive sources of information.